Effects of feedback delay on the performance of the transfer-controlled procedure in controlling CCS network overloads

Smith, D.E.
Bellcore, Red Bank, NJ, USA;

This paper appears in: **Selected Areas in Communications, IEEE Journal on**

Publication Date: April 1994
On page(s): 424 - 432
Volume: 12 , Issue: 3
ISSN: 0733-8716
Reference Cited: 7
CODEN: ISACEM
Inspec Accession Number: 4678935

Abstract:
This paper studies the performance of the common channel signaling link congestion control procedure in the presence of network latency (the delay between the onset of congestion and the reaction of traffic sources to congestion **information**). The context of the study is a focused overload on a signaling point. The results show that the procedure tends to synchronize the intervals of time when traffic sources may or may not transmit and it tends to overcontrol traffic. Consequently, call (not just message) throughput

suffers. The paper proposes and analyzes several partial remedies that are easy to implement. In particular, it recommends altering **timer** values and queue congestion **thresholds** and finds that the call throughput improves dramatically

Index Terms:

delays   queueing theory   telecommunication signalling   telecommunication traffic   telecommunications control   CCS network overloads control   call throughput   common channel signaling link   congestion control   congestion **information**   feedback delay   network latency   queue congestion **thresholds**   signaling point overload   **timer** values   traffic sources   transfer-controlled procedure

Documents that cite this document

Select link to view other documents in the database that cite this one.

Search Results   [PDF FULL-TEXT 816 KB]   PREV   NEXT   DOWNLOAD CITATION

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Search Results [PDF FULL-TEXT 600 KB] NEXT DOWNLOAD CITATION

Time threshold dimensioning and overload control in FDDI networks

Tangemann, M.
Inst. of Commun. Switching & **Data** Tech., Stuttgart Univ., Germany;

This paper appears in: **INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE**

Meeting Date: 05/04/1992 - 05/08/1992
Publication Date: 4-8 May 1992
Location: Florence Italy
On page(s): 363 - 371 vol.1
Reference Cited: 26
Inspec Accession Number: 4347766

Abstract:
The author addresses the problem of selecting **timer thresholds** for the asynchronous priorities in a Fiber Distributed **Data** Interface (FDDI) system. Based on an approximate analysis for the throughput under overload, two methods are derived. The first one is designed to meet requirements for the ratios between throughputs of stations with different priorities, whereas the second method includes absolute throughput values. Additionally, a proposition for dimensioning the **timer thresholds** of traffic classes that are deferred under overload is presented. Some examples for the application of these

methods to overload control in FDDI systems are given. The results indicate that FDDI's timed token protocol provides the possibility to guarantee bandwidth not only for synchronous traffic, but also for asynchronous traffic. The allocation of guaranteed bandwidth can be controlled easily by selecting the **timer thresholds** according to the rules derived

Index Terms:
FDDI protocols telecommunication traffic FDDI networks Fiber Distributed **Data** Interface approximate analysis asynchronous priorities asynchronous traffic deferred traffic classes guaranteed bandwidth overload control synchronous traffic throughput timed token protocol **timer thresholds**

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Refine Search

Search Results -

Terms	Documents
L1 and (interrupt near10 (threshold or maximum))	29

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L2

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Wednesday, March 17, 2004 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=USPT; PLUR=YES; OP=OR		
L2	l1 and (interrupt near10 (threshold or maximum))	29	L2
L1	((packet or data or information) near5 timer) same (threshold or maximum)	819	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L2	0

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L3

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Wednesday, March 17, 2004 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
DB=USOC,EPAB,JPAB,DWPI; PLUR=YES; OP=OR			
L3	L2	0	L3
DB=USPT; PLUR=YES; OP=OR			
L2	l1 and (interrupt near10 (threshold or maximum))	29	L2
L1	((packet or data or information) near5 timer) same (threshold or maximum)	819	L1

END OF SEARCH HISTORY

BEST AVAILABLE COPY

Refine Search

Search Results -

Terms	Documents
L2	0

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L4

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Wednesday, March 17, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=TDBD; PLUR=YES; OP=OR			
L4	L2	0	L4
DB=USOC,EPAB,JPAB,DWPI; PLUR=YES; OP=OR			
L3	L2	0	L3
DB=USPT; PLUR=YES; OP=OR			
L2	l1 and (interrupt near10 (threshold or maximum))	29	L2
L1	((packet or data or information) near5 timer) same (threshold or maximum)	819	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L1 and (timer same interrupt same threshold)	2

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L3

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Wednesday, March 17, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L3	L1 and (timer same interrupt same threshold)	2	L3
L2	L1 and (timer same interrupt)	4	L2
L1	5708817.pn. or 5717932.pn. or 6115776.pn. or 6195725.pn. or 6256660.pn.	5	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
(370/402 |370/412 |370/912 |709/235 |709/250 |709/231 |709/220 |709/301 |710/260 |710/261 |710/266 |710/310 |710/48 |710/62 |713/502).ccls.	5532

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L1

Refine Search

Recall Text   Clear   Interrupt

Search History

DATE: Wednesday, March 17, 2004   Printable Copy   Create Case

Set Name side by side	**Query**	**Hit Count**	**Se Nan** resu set
	DB=USPT; PLUR=YES; OP=OR		
L1	710/260,261,266,310,48,62;709/235,250,231,220,301;370/402,412,912;713/502.ccls.	5532	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L1 and L3	5

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L4

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Wednesday, March 17, 2004 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Se Nan** resu set
	DB=USPT; PLUR=YES; OP=OR		
L4	l1 and L3	5	L4
L3	l2 and (interrupt near10 (threshold or maximum))	29	L3
L2	((packet or data or information) near5 timer) same (threshold or maximum)	819	L2
L1	710/260,261,266,310,48,62;709/235,250,231,220,301;370/402,412,912;713/502.ccls.	5532	L1

END OF SEARCH HISTORY

EAST - [Untitled1:1]

File View Edit Tools Window Help

- Drafts
- Pending
- Active
  - L1: (338) ((packet or data c
  - L2: (10) l1 and (interrupt near10 threshold)
- Failed
- Saved
- Favorites
- Tagged (0)
- UDC
- Queue
- Trash

Search | List | Browse | Queue | Clear

DBs: USPAT

Plurals

Default operator: OR

Highlight all hit terms initially

BRS I... | IS&R ... | Image | Text | HTML

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments	Error Definition	Err
1	BRS	L1	338	((packet or data or information) near5 timer)	USPAT	2004/03/17 13:29			0
2	BRS	L2	10	l1 and (interrupt near10 threshold)	USPAT	2004/03/17 13:30			0

Start | Client Manager | EAST - [Untitled1...

BEST AVAILABLE COPY

EAST - [Untitled1:1]

File View Edit Tools Window Help

Drafts
Pending
Active
L1: (338) ((packet or data c
L2: (10) l1 and (interrupt r
Failed
Saved
Favorites
Tagged (0)
UDC
Queue
Trash

Search | List | Browse | Queue | Clear

DBs USPAT

Plurals

Default operator: OR

Highlight all hit terms initially

l1 and (interrupt near10 threshold)

BRS I... | IS&R ... | Image | Text | HTML

	U	1	Document ID	Issue Date	Pages	Title	Current OR	Current XRef	R
1	☐	☐	US 6476854 B1	20021105	32	Video eavesdropping and reverse assembly to transmit	348/143		
2	☐	☐	US 6467008 B1	20021015	23	Method and apparatus for indicating an interrupt in a	710/261		
3	☐	☐	US 6434651 B1	20020813	25	Method and apparatus for suppressing interrupts in a	710/260	709/235	
4	☐	☐	US 6278959 B1	20010821	25	Method and system for monitoring the performance	702/186	702/182; 711/117;	
5	☐	☐	US 6216182 B1	20010410	6	Method and apparatus for serving data with adaptable	710/48	709/233; 710/33;	
6	☐	☐	US 5789690 A	19980804	37	Electronic sound source having reduced spurious	84/633	84/604; 84/605;	
7	☐	☐	US 5767430 A	19980616	14	Sound source controlling device	84/602	84/604; 84/615	
8	☐	☐	US 5664097 A	19970902	9	System for delaying the activation of inactivity	713/200	713/189; 713/202	
9	☐	☐	US 5634015 A	19970527	40	Generic high bandwidth adapter providing data	710/310	370/402; 370/412;	
10	☐	☐	US 5367643 A	19941122	48	Generic high bandwidth adapter having data packet	710/62	370/412; 709/231	

Start | Client Manager | EAST - [Untitled1...

BEST AVAILABLE COPY

Your search matched **8** of **1013964** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevanc** **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.
(packet or information or data) and timer and thresh Search
☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Time threshold dimensioning and overload control in FDDI network**
Tangemann, M.;
INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE , 4-8 May 1992
Pages:363 - 371 vol.1

[Abstract] [PDF Full-Text (600 KB)] **IEEE CNF**

2 **A mobility management strategy for GPRS**
Yi-Bing Lin; Shun-Ren Yang;
Wireless Communications, IEEE Transactions on , Volume: 2 , Issue: 6 , Nov.

Pages:1178 - 1188

[Abstract] [PDF Full-Text (800 KB)] **IEEE JNL**

3 **A dynamic anchor-cell assisted paging with an optimal timer for PC networks**
Yang Xiao;
Communications Letters, IEEE , Volume: 7 , Issue: 8 , Aug. 2003
Pages:358 - 360

[Abstract] [PDF Full-Text (253 KB)] **IEEE JNL**

4 **Effects of feedback delay on the performance of the transfer-contro pr cedure in c ntr lling CCS netw rk verloads**
Smith, D.E.;
Selected Areas in Communications, IEEE Journal on , Volume: 12 , Issue: 3 , 1994
Pages:424 - 432

[Abstract] [PDF Full-Text (816 KB)] **IEEE JNL**

---

5 **A c mparative study f QoS r uting schemes that t lerate impreci state inf rmati n**
Xin Yuan; Wei Zheng; Shiling Ding;
Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on , 14-16 Oct. 2002
Pages:230 - 235

[Abstract] [PDF Full-Text (282 KB)] **IEEE CNF**

---

6 **Scalability of routing advertisement for QoS routing in an IP netwo with guaranteed QoS**
Yu-Kung Ke; Copeland, J.A.;
Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE , Volum 1 , 27 Nov.-1 Dec. 2000
Pages:605 - 610 vol.1

[Abstract] [PDF Full-Text (536 KB)] **IEEE CNF**

---

7 **Performance of TCP traffic and ATM feedback congestion control mechanisms**
Iliadis, I.;
Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communicat The Key to Global Prosperity , Volume: 3 , 18-22 Nov. 1996
Pages:1930 - 1934 vol.3

[Abstract] [PDF Full-Text (492 KB)] **IEEE CNF**

---

8 **Macromodel of timer for electrical level simulators**
Peic, R.V.;
Electrotechnical Conference, 1991. Proceedings., 6th Mediterranean , 22-24 1991
Pages:125 - 128 vol.1

[Abstract] [PDF Full-Text (240 KB)] **IEEE CNF**

Time threshold dimensioning and overload control in FDDI networks

Tangemann, M.
Inst. of Commun. Switching & **Data** Tech., Stuttgart Univ., Germany;

This paper appears in: **INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE**

Meeting Date: 05/04/1992 - 05/08/1992
Publication Date: 4-8 May 1992
Location: Florence Italy
On page(s): 363 - 371 vol.1
Reference Cited: 26
Inspec Accession Number: 4347766

Abstract:
The author addresses the problem of selecting **timer thresholds** for the asynchronous priorities in a Fiber Distributed **Data** Interface (FDDI) system. Based on an approximate analysis for the throughput under overload, two methods are derived. The first one is designed to meet requirements for the ratios between throughputs of stations with different priorities, whereas the second method includes absolute throughput values. Additionally, a proposition for dimensioning the **timer thresholds** of traffic classes that are deferred under overload is presented. Some examples for the application of these

methods to overload control in FDDI systems are given. The results indicate that FDDI's timed token protocol provides the possibility to guarantee bandwidth not only for synchronous traffic, but also for asynchronous traffic. The allocation of guaranteed bandwidth can be controlled easily by selecting the **timer thresholds** according to the rules derived

Index Terms:
FDDI protocols telecommunication traffic FDDI networks Fiber Distributed **Data** Interface approximate analysis asynchronous priorities asynchronous traffic deferred traffic classes guaranteed bandwidth overload control synchronous traffic throughput timed token protocol **timer thresholds**

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Effects of feedback delay on the performance of the transfer-controlled procedure in controlling CCS network overloads

Smith, D.E.
Bellcore, Red Bank, NJ, USA;

This paper appears in: **Selected Areas in Communications, IEEE Journal on**

Publication Date: April 1994
On page(s): 424 - 432
Volume: 12 , Issue: 3
ISSN: 0733-8716
Reference Cited: 7
CODEN: ISACEM
Inspec Accession Number: 4678935

Abstract:
This paper studies the performance of the common channel signaling link congestion control procedure in the presence of network latency (the delay between the onset of congestion and the reaction of traffic sources to congestion **information**). The context of the study is a focused overload on a signaling point. The results show that the procedure tends to synchronize the intervals of time when traffic sources may or may not transmit and it tends to overcontrol traffic. Consequently, call (not just message) throughput

suffers. The paper proposes and analyzes several partial remedies that are easy to implement. In particular, it recommends altering **timer** values and queue congestion **thresholds** and finds that the call throughput improves dramatically

Index Terms:
delays queueing theory telecommunication signalling telecommunication traffic telecommunications control CCS network overloads control call throughput common channel signaling link congestion control congestion **information** feedback delay network latency queue congestion **thresholds** signaling point overload **timer** values traffic sources transfer-controlled procedure

Documents that cite this document
Select link to view other documents in the database that cite this one.

First Hit Fwd Refs

Generate Collection Print

L2: Entry 2 of 29 File: USPT Nov 5, 2002

DOCUMENT-IDENTIFIER: US 6476854 B1
TITLE: Video eavesdropping and reverse assembly to transmit video action to a remote console

Brief Summary Text (11):
The present invention disclosed and claimed herein includes a method and apparatus for monitoring video activity, reverse assembling the video transactions and transmitting the video transactions from a remote computer to a local computer. The local computer has a modem and conventional communications software for communicating with the remote computer of the preferred embodiment. The preferred embodiment includes hardware and software components, but other configurations are contemplated. An integrated remote console (IRC) device is provided in the remote computer for snooping video transactions passed on a bus from a processor to a video controller and for compressing the video transactions into an encoded format. Data and address comparison logic is provided in the IRC device to encode block move operations and fill operations (repetitive character strings). The data is encoded into a proprietary data stream which includes addressing information. A first in first out (FIFO) buffer accumulates the data until a threshold is reached and a system management interrupt (SMI) is indicated.

Detailed Description Text (28):
Also, once the call is determined to be for the IRC subsystem 180, video is provided via the telephone line to the local computer so that the operator may view everything that an operator physically present at the remote computer C could view. The IRC device 156 performs snooping on all video activity to non-intrusively capture video data, command and control information as it passes by on the PCI bus 114, as shown as path 2. The IRC device 156 then compresses or encodes the information into a proprietary data stream which is read from the IRC device 156 by the virtual processor 101v when a certain activity threshold is reached, or a staleness timer causes the flushing of video data from the IRC device 156. The virtual processor 101v further compresses the video information as it is translated into an ANSI (American National Standards Institute) character format for transmitting to the local computer 190. The local computer 190 displays the received information on its video monitor when received.

Detailed Description Text (39):
A data conversion module 224 (CONVERT) receives the processed address, data and control information from the STG345 module 222 where the CONVERT module 222 converts the information flow from a 32-bit bus into a 64-bit bus with byte lane enables. Concatenation is performed by disregarding empty byte lanes and by shifting. The 64-bit bus is received into a first-in-first-out (FIFO) memory module 226 and stored until read out by the SMI firmware 1000. The FIFO module 226 includes certain preprocessing elements that cause all byte lanes to be filled before incrementing the FIFO input pointer. The FIFO 226 of the preferred embodiment is 8 quadwords deep. When a programmable FIFO threshold is reached, a system management interrupt causes the processor to enter system management mode and empty the FIFO 226. The processing performed by the SMI firmware 1000 is discussed below.

First Hit Fwd Refs

Generate Collection Print

L2: Entry 2 of 29 File: USPT Nov 5, 2002

US-PAT-NO: 6476854
DOCUMENT-IDENTIFIER: US 6476854 B1

TITLE: Video eavesdropping and reverse assembly to transmit video action to a remote console

DATE-ISSUED: November 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Emerson; Theodore F.	Houston	TX		
Michaels; Peter J.	Plano	TX		
Krontz; Jeoff	Houston	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Compaq Information Technologies Group, L.P.	Houston	TX			02

APPL-NO: 08/ 733254 [PALM]
DATE FILED: October 18, 1996

INT-CL: [07] H04 N 7/18

US-CL-ISSUED: 348/143
US-CL-CURRENT: 348/143

FIELD-OF-SEARCH: 348/15, 348/16, 348/143, 348/150-155, 348/169

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5111288	May 1992	Blachshear	348/143
5202759	April 1993	Laycock	348/152
5218627	June 1993	Corey et al.	348/16
5521634	May 1996	McGary	348/169
5598209	January 1997	Cortjens et al.	348/15

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
59-93675	December 1984	JP	

ART-UNIT: 2613

PRIMARY-EXAMINER: Lee; Richard

ATTY-AGENT-FIRM: Akin, Gump, Strauss, Hauer & Feld, LLP

ABSTRACT:

A computer system having integrated remote console functionality. Cycles intended for a video graphics controller are snooped to acquire the video data or control information contained in the cycle. Analysis is performed on the video data to detect sequential or repetitive operations. The video data is encoded into higher level primitives, if possible. The video data and primitives are held in a first-in-first-out (FIFO) memory until the FIFO reaches a critical level, or a staleness timer times out. Special firmware executed in system management mode reads the FIFO and converts the video data and primitives into conventional ASCII text or the required format. The firmware also maintains a copy of the video frame buffer to further encode the video data, if possible. The firmware then transmits the conventional ASCII text via a modem to a user stationed at a remote computer system.

29 Claims, 20 Drawing figures

L2: Entry 3 of 29 File: USPT Oct 15, 2002

DOCUMENT-IDENTIFIER: US 6467008 B1
**** See image for Certificate of Correction ****
TITLE: Method and apparatus for indicating an interrupt in a network interface

Brief Summary Text (12):
In one embodiment of the invention a system and method for polling a network interface are provided. In this embodiment an interrupt that would normally alert a host processor to the arrival of a network packet is suspended during a polling mode of operation. Each time the network interface is polled, any waiting packets are processed. If a threshold amount of time or a threshold number of packets are received without being processed, however, interrupts may be enabled to ensure the packets are serviced. Thus, a polling mode of operation may be combined with interrupt modulation in one embodiment of the invention.

Brief Summary Text (15):
Each time the network interface is polled, and upon the completion of processing of a interrupt by a host processor, a time counter and/or a packet counter are initialized. Illustratively, the counters are set to programmable threshold values representing a maximum period of time that is allowed to pass and a maximum number of packets that may be received before issuing another interrupt. After initialization, the counters begin decrementing toward final programmable values (e.g., zero). As long as polling continues in this embodiment, the counters will be repeatedly re-initialized and will therefore never expire and no interrupts will be generated. In an alternative embodiment the counters are initialized to initial values and thereafter increment toward threshold values.

Detailed Description Text (22):
Illustratively, the time counter and packet counter are set to threshold values after a host computer processes an interrupt. Suitable threshold values, which may be stored in programmable registers or other data structures, are twenty microseconds of time and seven packets for a moderate level of traffic. Thereafter, the time counter decrements (e.g., counts down toward zero) in response to a clock signal or other means of noting the passage of time, and the packet counter decrements in response to a packet transfer signal or other indicator that a packet was transferred to the host computer. A packet transfer signal may, for example, be generated by a NIC module responsible for copying a packet to host memory. Until either of the time counter or packet counter reaches zero or some other final value, which may be stored in a register or other programmable data storage unit, an interrupt will not be generated in response to the transfer of a packet. After a final value is reached, an interrupt may be generated for a packet transferred after the last interrupt was processed by the host computer.

Detailed Description Text (25):
Interrupt modulation according to the present invention differs from that of the '758 patent in how the time and packet counters are reset. In particular, rather than being reset at the time the interrupt is issued, in a present embodiment of the invention the counters are not reset until after the host computer processes an interrupt. As described above, in one embodiment of the invention a second interrupt is not initiated by a network interface until a minimum period of time passes or a specified number of packets are received after a host processor

completes servicing a first interrupt. One skilled in the art will recognize that the present scheme provides additional time separation between successive interrupts and thus further decreases the amount of host processor time expended in servicing interrupts from the network interface. Under the '758 patent, if a host processor requires nearly a full threshold of time (e.g., twenty microseconds) to finish its processing of one interrupt, interrupts may be enabled and an interrupt issued shortly after the first one is serviced.

Detailed Description Text (26):
FIG. 1 is a diagram of an interrupt modulator for a NIC according to one embodiment of the invention. In this embodiment time and packet counters are set to their threshold values (e.g. twenty microseconds and seven packets, respectively) after an interrupt is processed by a host computer and then decremented in accordance with the passage of time and the transfer of packets. A counter expires when it reaches a final value (e.g., zero), at which time another interrupt may be generated. Although threshold values of approximately twenty microseconds and approximately seven packets are employed in embodiments of the invention discussed below, in alternative embodiments a wide range of thresholds will be suitable. In particular, one network environment in which an embodiment of the invention may be practiced (e.g., such as the Internet) employs Ethernet, IP and TCP protocols, respectively, at layers two, three and four of an associated protocol stack. In this environment a range of twenty to fifty microseconds may be appropriate for a time counter and a range of four to ten packets may be appropriate for a packet counter.

Detailed Description Text (30):
The time threshold registers and packet threshold registers depicted in FIG. 1 are implemented in programmable memory (e.g., register, RAM, flash memory). Thus, they may be set or modified by software (e.g., a device driver) operating on a host computer. As depicted in FIG. 1, multiple threshold registers may be coupled to a time and/or packet counter in order to allow the counters to be reset to different values depending on the level of network traffic. In particular, different threshold values may be desirable depending on the level of traffic received at the NIC or processed by a host processor. The time and packet counters may therefore be re-initialized to different thresholds as the traffic fluctuates. An interrupt modulator is not limited to a particular number of threshold registers and may include any number of them greater than or equal to one. In another alternative embodiment, threshold time and/or packet counts are stored in read-only memory.

Detailed Description Text (36):
When initialized, time counter 102 is set to the threshold value stored in time threshold register 104a (e.g., or an alternate time threshold register such as register 104b). As the time counter is decremented in response to clock signal 110, it is compared to the value in final time register 116 by comparator 118. If time counter 102 is decremented to or beyond the value in final time register 116 (e.g., indicating the passage of at least twenty microseconds of time), then time expired signal 120 is activated. The value stored in time threshold register 104a may be modified (e.g., increased) as the rate of receipt of network traffic increases. Or, as shown in FIG. 1, a different threshold value may be loaded from a different time threshold register depending upon the level of traffic (e.g., number of packets processed in an interrupt, number of packets received in a given period of time). The determination of which time threshold register's value to use for initializing time counter 102 may be made on the basis of the amount of traffic being transferred to the host computer. The value in final time register 116 may also be altered.

Detailed Description Text (37):
Similarly, when packet counter 106 is initialized, it is set to the threshold value stored in packet threshold register 108a (or an alternate packet threshold register such as register 108b). As the packet counter is decremented in response to packet

transfer signal 112, it is compared to the value in final packet register 122 by comparator 124. If packet counter 106 is decremented to or beyond the value in final packet register 122, then packets exceeded signal 126 is activated. The value stored in packet threshold register 108a may be modified (e.g., increased) as the rate of receipt of network traffic increases. Or, as shown in FIG. 1, a different threshold value may be loaded from a different packet threshold register depending upon the level of traffic (e.g., number of packets processed in an interrupt, number of packets received in a given period of time). The value in final packet register 122 may also be altered.

Detailed Description Text (51):
In state 304, interrupts associated with the transfer of packets from a network interface are temporarily disabled. Certain steps toward the issuance of an interrupt may still be performed in response to the transfer of a packet, such as the setting of an indicator in a status register. As described below, however, no packet transfer interrupt will be transmitted to a host processor until either the time or packet counter reaches its threshold value.

Detailed Description Text (52):
In state 306, one or both of the time and packet counters decrement (e.g., increment negatively, or toward zero). In particular, the time counter will generally continuously decrement, from the time it is initialized, until it is re-initialized. It may, however, be halted or suspended (e.g., cease decrementing) when interrupts are enabled. After an interrupt is processed the time counter is re-initialized to a threshold value and once again begins decrementing. A packet counter, however, decrements in response to an event that is less certain than the passage of time--the transfer of a packet from the network interface. Illustratively, for each packet transferred by the network interface to a host computer the packet counter count decreases by one.

Detailed Description Text (60):
When packets are transferred from a network interface at a very high rate, a packet counter's threshold may be quickly surmounted each time it is reset. Each time the threshold is exceeded another interrupt would then be generated to a host processor. Of course the packet counter threshold may be set to a relatively high value in response (e.g., by loading a threshold value from a different packet threshold register), but if the transfer rate remains very high then each time the processor receives an interrupt it may expend an inordinate amount of time processing a large number of packets. Some communication protocols may not be able to adequately function in such an environment. Or, if the rate of packet transfer to the host computer decreases precipitously after the packet counter threshold is increased, then the next interrupt, and the subsequent processing of one or more packets, may be delayed.

Detailed Description Text (61):
Even the use of a time counter may not alleviate the problems associated with heavy traffic levels. In particular, if the time counter has a relatively high time threshold, the number of packets transferred by a network interface before the threshold is exceeded may again require the processor to spend a significant period of time processing the packets when an interrupt is generated. If the time threshold is set to a low value (e.g., by using a different time threshold register), then the processor may be over-burdened with interrupts--similar to the result of employing a low packet counter threshold.

Detailed Description Text (80):
From decrement counter state 410, polling state 406 may be entered because a polling timer expires or reaches its threshold, as represented by transition 416. A polling timer's threshold is preferably lower than the interrupt modulator's time threshold to reflect the preference for a polling operation over the generation of an interrupt. If, however, the time counter expires or reaches its final value

before a polling operation can be completed (after which the time counter is re-initialized), transition 418 illustrates the entry into interrupt enabled state 418 from decrement counter state 410. Transition 418 may also be initiated by the receipt of a threshold number of packets. From interrupt enabled state 404, initialize counter state 408 is entered after an interrupt is serviced by a host processor, as indicated by transition 420.

Detailed Description Text (103):
Illustratively, the status register is accessed through an alias register (e.g., alias address) during heavy levels of network traffic (e.g., when a polling mode of operation is active). Otherwise, the status register may be accessed directly. In addition, it was described above that different threshold values for time and/or packet counters in an interrupt modulator may be desirable for different levels of traffic. Therefore, in an embodiment of the invention in which an interrupt modulator stores multiple time and/or packet threshold values (e.g., in separate threshold registers), the threshold value that is loaded into a counter may be determined by how the status register is accessed.

Detailed Description Text (107):
In state 704, it is determined whether the level of traffic or rate of interrupts or some other traffic measure indicates that it would be more efficient to implement a polling mode of operation. It may, for example, be determined that the number of packets processed during an interrupt exceeds a programmable threshold (e.g., fifty). Other criteria for making this determination include the rate at which interrupts are issued by the network interface for packets transferred to the host (e.g., approximately 10,000 per second) or the level of processor utilization (e.g., approximately ninety percent). These thresholds may be measured on a one-time or instantaneous basis or may be averaged or otherwise combined over a period of time or a number of interrupts. The illustrated procedure returns to state 702 if an applicable threshold is not met.

Detailed Description Text (109):
In state 708 thresholds for one or more interrupt modulator counters (e.g., a time and/or packet counter) are set or increased from lower values used during interrupt modulation. The counters are activated and thereafter decrement or increment as described previously. By increasing the thresholds, any interrupts that may be issued by the interrupt modulator are issued with less frequency, if at all. Delaying interrupts allows the polling module to perform its polling and packet processing and allows a host processor to avoid the overhead involved in servicing an interrupt.

Detailed Description Text (110):
In this embodiment an interrupt modulator may store multiple threshold values in multiple registers for a packet and/or a time counter in order to facilitate the modification of counter thresholds. With multiple threshold registers, the counters may be easily re-initialized or switched to use different thresholds. In particular, when operating in a moderate level of network traffic (e.g., without using an alias register) a packet or time counter may be re-initialized to a first threshold every time a status register is cleared (e.g., because of an interrupt). However, when operating in a heavier traffic environment characterized by the use of polling and, possibly, an alias register, a counter may be re-initialized to a second threshold when the alias register is read by the polling software.

Detailed Description Text (113):
State 710 demonstrates that if an interrupt modulator reaches a threshold value, the network interface will not continue waiting for a polling operation. In one alternative embodiment a polling timer may be examined more frequently than a packet or time counter. In other words, states 710-712 and 714-716 may be reversed, such that the interrupt modulator's counters are only examined when it is determined that it is not time for a polling operation.

Detailed Description Text (116):
During polling in a heavy traffic environment the software may have many packets to process each time it examines a network interface's status register or alias register. In order to prevent an interrupt modulator's counter from reaching its threshold during the processing of the packets, the polling software may temporarily pause, suspend or interrupt operation in order to re-initialize the counter(s). For example, if an interrupt modulator's time counter is set to one millisecond, the polling software will attempt to ensure that it re-initializes the time counter less than one millisecond after it was last re-initialized. The polling software may examine the time counter, its own counter, or another counter or clock signal in order to determine when it should re-initialize a counter.

Detailed Description Text (117):
Illustratively, the threshold for an interrupt modulator's time counter is determined, at least in part, by the amount of time that the polling software is expected to need in order to process an expected amount of traffic. The time counter threshold, and the frequency of polling, may also be affected by the speed of a host computer processor.

CLAIMS:

17. The interrupt modulator of claim 16, further comprising: a first time threshold storage device configured to store a first initial time count; a second time threshold storage device configured to store a second initial time count; a first packet threshold storage device configured to store a fist initial packet count; and a second packet threshold storage device configured to store a second initial packet count; packet wherein said time count is reset to one of said first initial time count and said second initial time count in response to said poll or a termination of processing of said interrupt; and wherein said packet count is reset to one of said first initial packet count and said second initial packet count in response to said poll or said termination of processing of said interrupt.

First Hit Fwd Refs

Generate Collection Print

L2: Entry 3 of 29 File: USPT Oct 15, 2002

US-PAT-NO: 6467008
DOCUMENT-IDENTIFIER: US 6467008 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for indicating an interrupt in a network interface

DATE-ISSUED: October 15, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gentry, Jr.; Denton E.	Fremont	CA		
Cheng; Linda T.	San Jose	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Sun Microsystems, Inc.	Santa Clara	CA			02

APPL-NO: 09/ 259936 [PALM]
DATE FILED: March 1, 1999

INT-CL: [07] G06 F 13/24

US-CL-ISSUED: 710/261
US-CL-CURRENT: 710/261

FIELD-OF-SEARCH: 710/260-269

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO.	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5319752	June 1994	Petersen et al.	395/250
☐	5414858	May 1995	Hoffman et al.	340/3.4
☐	5463752	October 1995	Benhase et al.	710/262
☐	5471618	November 1995	Isfeld	340/3.51
☐	5485584	January 1996	Hausmann et al.	395/842
☐	5797037	August 1998	Ecclesine	395/868
☐	6065073	May 2000	Booth	370/908

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 4 96 177	July 1992	EP	
0 752 799	January 1997	EP	
0 852 357	July 1998	EP	
2 282 474	April 1995	GB	

OTHER PUBLICATIONS

Pending U.S. patent application Ser. No. 09/259,445, entitled "Method and Apparatus for Distributing Network Processing on a Multiprocessor Computer," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3481-JTF).
Pending U.S. patent application Ser. No. 09/260,367, entitled "Method and Apparatus for Suppressing Interrupts in a High-Speed Network Environment", by Denton Gentry, filed Mar. 1, 1999 (Attorney Docket SUN-P3482-JTF).
Pending U.S. patent application Ser. No. 09/259,736, entitled "Method and Apparatus for Modulating Interrupts in a Network Interface," by Denton Gentry et al., filed Mar. 1, 1999 (Attorney Docket SN-P3483-JTF).
Pending U.S. patent application Ser. No. 09/259,765, entitled "A High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3485-JTF).
Pending U.S. patent application Ser. No. 09/260,618, entitled "Method and Apparatus for Classifying Network Traffic in a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3486-JTF).
Pending U.S. patent application Ser. No. 09/259,932, entitled "Method and Apparatus for Managing a Network Flow in a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3487-JTF).
Pending U.S. patent Application Ser. No. 09/260,324, entitled "Method and Apparatus for Dynamic Packet Batching with a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3488-JTF).
Pending U.S. patent application Ser. No. 09/258,952, entitled "Method and Apparatus for Early Random Discard of Packets," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3490-JTF).
Pending U.S. patent application Ser. No. 09/260,333, entitled "Method and Apparatus for Data Re-Assembly with a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3570-JTF).
Pending U.S. patent application Ser. No. 08/258,955, entitled "Dynamic Parsing in a High Performance Network Interface," by Denton Gentry, filed Mar. 1, 1999 (Attorney Docket SUN-P3715-JTF).

ART-UNIT: 2181

PRIMARY-EXAMINER: Lefkowitz; Sumati

ATTY-AGENT-FIRM: Park, Vaughan & Fleming LLP

ABSTRACT:

A network interface is polled by a process operating on a host computer system. Each time the network interface is polled the process determines whether any packets have been transferred to the host. If so, they are processed. Interrupts that would normally be issued from the network interface in response to the transfer of packets are suppressed or postponed during the polling mode of operation. If, however, a predetermined period of time has elapsed or a predetermined number of packets have been transferred since a previous poll or a

previous interrupt, then an interrupt may be generated.

21 Claims, 7 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L2: Entry 4 of 29 File: USPT Aug 13, 2002

DOCUMENT-IDENTIFIER: US 6434651 B1
TITLE: Method and apparatus for suppressing interrupts in a high-speed network environment

Abstract Text (1):
A network interface is polled by a process operating on a host computer system. Each time the network interface is polled, the process determines whether any packets have been received. If so, they are processed. Interrupts that would normally be issued by the network interface in response to the transfer of packets to the host system are suppressed or postponed during the polling mode of operation. If, however, a predetermined period of time has elapsed or a predetermined number of packets have been received since a previous poll or a previous interrupt, then an interrupt may be generated. The rate at which interrupts may be issued is modulated to decrease the interrupt-processing burden placed on the processor. A time counter may be used to track the passage of time and a packet counter may be used to track the number of packets transferred by the network interface. After each polling operation or processing of an interrupt by the host processor, the time and packet counters are reset to threshold values and thereafter begin decrementing toward a final time count and a final packet count, respectively. Thus, a packet transferred after one polling operation or interrupt does not cause the issuance of an interrupt to the host processor unless a time or packet counter reaches its final value (e.g. zero). The threshold time count and packet count may be adjusted to ensure that interrupts are generated often enough to avoid a negative impact on the processing of packets if, for example, the polling process is blocked.

Brief Summary Text (12):
In one embodiment of the invention, a system and method for polling a network interface are provided. In this embodiment an interrupt that would normally alert a host processor to the arrival of a network packet is suspended during a polling mode of operation. Each time the network interface is polled, any waiting packets are processed. However, if a threshold amount of time passes, or a threshold number of packets are received without being processed, interrupts may be enabled to ensure the packets are serviced. Thus, a polling mode of operation may be combined with interrupt modulation in one embodiment of the invention.

Brief Summary Text (15):
Each time the network interface is polled, and upon the completion of processing of an interrupt by a host processor, a time counter and/or a packet counter are initialized. Illustratively, the counters are set to programmable threshold values representing, respectively, a maximum period of time that is allowed to pass and a maximum number of packets that may be received before issuing another interrupt. After initialization, the counters begin decrementing toward final programmable values (e.g., zero). As long as polling continues in this embodiment, the counters will be repeatedly re-initialized and will therefore never expire and no interrupts will be generated. In an alternative embodiment, the counters are initialized to initial values and thereafter increment toward threshold values.

Detailed Description Text (22):
Illustratively, the time counter and packet counter are set to threshold values

after a host computer processes an interrupt. Suitable threshold values, which may be stored in programmable registers or other data structures, are twenty microseconds of time and seven packets for a moderate level of traffic. Thereafter, the time counter decrements (e.g., counts down toward zero) in response to a clock signal or other means of noting the passage of time, and the packet counter decrements in response to a packet transfer signal or other indicator that a packet was transferred to the host computer. For example, a packet transfer signal may be generated by a NIC module responsible for copying a packet to host memory. Until either the time counter or packet counter reaches zero or some other final value, which may be stored in a register or other programmable data storage unit, an interrupt will not be generated in response to the transfer of a packet. After a final value is reached, an interrupt may be generated for a packet transferred after the last interrupt,was processed by the host computer.

Detailed Description Text (25):
Interrupt modulation according to the present invention differs from that of the '758 patent in how the time and packet counters are reset. In particular, rather than being reset at the time the interrupt is issued, in a present embodiment of the invention the counters are not reset until after the host computer processes an interrupt. As described above, in one embodiment of the invention a second interrupt is not initiated by a network interface until a minimum period of time passes or a specified number of packets are received after a host processor completes servicing a first interrupt. One skilled in the art will recognize that the present scheme provides additional time separation between successive interrupts and thus further decreases the amount of host processor time expended in servicing interrupts from the network interface. Under the '758 patent, if a host processor requires nearly a full threshold of time (e.g., twenty microseconds) to finish its processing of one interrupt, interrupts may be enabled and an interrupt issued shortly after the first one is serviced.

Detailed Description Text (26):
FIG. 1 is a diagram of an interrupt modulator for a NIC according to one embodiment of the invention. In this embodiment time and packet counters are set to their threshold values (e.g. twenty microseconds and seven packets, respectively) after an interrupt is processed by a host computer and then decremented in accordance with the passage of time and the transfer of packets. A counter expires when it reaches a final value (e.g., zero), at which time another interrupt may be generated. Although threshold values of approximately twenty microseconds and approximately seven packets are employed in embodiments of the invention discussed below, in alternative embodiments a wide range of thresholds will be suitable. In particular, one network environment in which an embodiment of the invention may be practiced (e.g., such as the Internet) employs Ethernet, IP and TCP protocols, respectively, at layers two, three and four of an associated protocol stack. In this environment a range of twenty to fifty microseconds may be appropriate for a time counter and a range of four to ten packets may be appropriate for a packet counter.

Detailed Description Text (30):
The time threshold registers and packet threshold registers depicted in FIG. 1 are implemented in programmable memory (e.g., register, RAM, flash memory). Thus, they may be set or modified by software (e.g., a device driver) operating on a host computer. As depicted in FIG. 1, multiple threshold registers may be coupled to a time and/or packet counter in order to allow the counters to be reset to different values depending on the level of network traffic. In particular, different threshold values may be desirable depending on the level of traffic received at the NIC or processed by a host processor. The time and packet counters may therefore be re-initialized to different thresholds as the traffic fluctuates. An interrupt modulator is not limited to a particular number of threshold registers and may include any number of them greater than or equal to one. In another alternative embodiment, threshold time and/or packet counts are stored in read-only memory.

Detailed Description Text (36):
When initialized, time counter 102 is set to the threshold value stored in time threshold register 104a (e.g., or an alternate time threshold register such as register 104b). As the time counter is decremented in response to clock signal 110, it is compared to the value in final time register 116 by comparator 118. If time counter 102 is decremented to or beyond the value in final time register 116 (e.g., indicating the passage of at least twenty microseconds of time), then time expired signal 120 is activated. The value stored in time threshold register 104a may be modified (e.g., increased) as the rate of receipt of network traffic increases. Or, as shown in FIG. 1, a different threshold value may be loaded from a different time threshold register depending upon the level of traffic (e.g., number of packets processed in an interrupt, number of packets received in a given period of time). The determination of which time threshold register's value to use for initializing time counter 102 may be made on the basis of the amount of traffic being transferred to the host computer. The value in final time register 116 may also be altered.

Detailed Description Text (37):
Similarly, when packet counter 106 is initialized, it is set to the threshold value stored in packet threshold register 108a (or an alternate packet threshold register such as register 108b). As the packet counter is decremented in response to packet transfer signal 112, it is compared to the value in final packet register 122 by comparator 124. If packet counter 106 is decremented to or beyond the value in final packet register 122, then packets exceeded signal 126 is activated. The value stored in packet threshold register 108a may be modified (e.g., increased) as the rate of receipt of network traffic increases. Or, as shown in FIG. 1, a different threshold value may be loaded from a different packet threshold register depending upon the level of traffic (e.g., number of packets processed in an interrupt, number of packets received in a given period of time). The value in final packet register 122 may also be altered.

Detailed Description Text (51):
In state 304, interrupts associated with the transfer of packets from a network interface are temporarily disabled. Certain steps toward the issuance of an interrupt may still be performed in response to the transfer of a packet, such as the setting of an indicator in a status register. As described below, however, no packet transfer interrupt will be transmitted to a host processor until either the time or packet counter reaches its threshold value.

Detailed Description Text (52):
In state 306, one or both of the time and packet counters decrement (e.g., increment negatively, or toward zero). In particular, the time counter will generally continuously decrement, from the time it is initialized, until it is re-initialized. It may, however, be halted or suspended (e.g., cease decrementing) when interrupts are enabled. After an interrupt is processed the time counter is re-initialized to a threshold value and once again begins decrementing. A packet counter, however, decrements in response to an event that is less certain than the passage of time--the transfer of a packet from the network interface. Illustratively, for each packet transferred by the network interface to a host computer the packet counter count decreases by one.

Detailed Description Text (60):
When packets are transferred from a network interface at a very high rate, a packet counter's threshold may be quickly surmounted each time it is reset. Each time the threshold is exceeded another interrupt would then be generated to a host processor. Of course the packet counter threshold may be set to a relatively high value in response (e.g., by loading a threshold value from a different packet threshold register), but if the transfer rate remains very high then each time the processor receives an interrupt it may expend an inordinate amount of time

processing a large number of packets. Some communication protocols may not be able to adequately function in such an environment. Or, if the rate of packet transfer to the host computer decreases precipitously after the packet counter threshold is increased, then the next interrupt, and the subsequent processing of one or more packets, may be delayed.

Detailed Description Text (61):
Even the use of a time counter may not alleviate the problems associated with heavy traffic levels. In particular, if the time counter has a relatively high time threshold, the number of packets transferred by a network interface before the threshold is exceeded may again require the processor to spend a significant period of time processing the packets when an interrupt is generated. If the time threshold is set to a low value (e.g., by using a different time threshold register), then the processor may be over-burdened with interrupts--similar to the result of employing a low packet counter threshold.

Detailed Description Text (80):
From decrement counter state 410, polling state 406 may be entered because a polling timer expires or reaches its threshold, as represented by transition 416. A polling timer's threshold is preferably lower than the interrupt modulator's time threshold to reflect the preference for a polling operation over the generation of an interrupt. If, however, the time counter expires or reaches its final value before a polling operation can be completed (after which the time counter is re-initialized), transition 418 illustrates the entry into interrupt enabled state 404 from decrement counter state 410. Transition 418 may also be initiated by the receipt of a threshold number of packets. From interrupt enabled state 404, initialize counter state 408 is entered after an interrupt is serviced by a host processor, as indicated by transition 420.

Detailed Description Text (103):
Illustratively, the status register is accessed through an alias register (e.g., alias address) during heavy levels of network traffic (e.g., when a polling mode of operation is active). Otherwise, the status register may be accessed directly. In addition, it was described above that different threshold values for time and/or packet counters in an interrupt modulator may be desirable for different levels of traffic. Therefore, in an embodiment of the invention in which an interrupt modulator stores multiple time and/or packet threshold values (e.g., in separate threshold registers), the threshold value that is loaded into a counter may be determined by how the status register is accessed.

Detailed Description Text (107):
In state 704, it is determined whether the level of traffic or rate of interrupts or some other traffic measure indicates that it would be more efficient to implement a polling mode of operation. For example, it may be determined that the number of packets processed during an interrupt exceeds a programmable threshold (e.g., fifty). Other criteria for making this determination include the rate at which interrupts are issued by the network interface for packets transferred to the host (e.g., approximately 10,000 per second) or the level of processor utilization (e.g., approximately ninety percent). These thresholds may be measured on a one-time or instantaneous basis, or may be averaged or otherwise combined over a period of time or a number of interrupts. The illustrated procedure returns to state 702 if an applicable threshold is not met.

Detailed Description Text (109):
In state 708 thresholds for one or more interrupt modulator counters (e.g., a time and/or packet counter) are set or increased from lower values used during interrupt modulation. The counters are activated and thereafter decrement or increment as described previously. By increasing the thresholds, any interrupts that may be issued by the interrupt modulator are issued with less frequency, if at all. Delaying interrupts allows the polling module to perform its polling and packet

processing and allows a host processor to avoid the overhead involved in servicing an interrupt.

Detailed Description Text (110):
In this embodiment an interrupt modulator may store multiple threshold values in multiple registers for a packet and/or a time counter in order to facilitate the modification of counter thresholds. With multiple threshold registers, the counters may be easily re-initialized or switched to use different thresholds. In particular, when operating in a moderate level of network traffic (e.g., without using an alias register) a packet or time counter may be re-initialized to a first threshold every time a status register is cleared (e.g., because of an interrupt). However, when operating in a heavier traffic environment characterized by the use of polling and, possibly, an alias register, a counter may be re-initialized to a second threshold when the alias register is read by the polling software.

Detailed Description Text (113):
State 710 demonstrates that if an interrupt modulator reaches a threshold value, the network interface will not continue waiting for a polling operation. In one alternative embodiment a polling timer may be examined more frequently than a packet or time counter. In other words, states 710-712 and 714-716 may be reversed, such that the interrupt modulator's counters are only examined when it is determined that it is not time for a polling operation.

Detailed Description Text (116):
During polling in a heavy traffic environment the software may have many packets to process each time it examines a network interface's status register or alias register. In order to prevent an interrupt modulator's counter from reaching its threshold during the processing of the packets, the polling software may temporarily pause, suspend or interrupt operation in order to re-initialize the counter(s). For example, if an interrupt modulator's time counter is set to one millisecond, the polling software will attempt to ensure that it re-initializes the time counter less than one millisecond after it was last re-initialized. The polling software may examine the time counter, its own counter, or another counter or clock signal in order to determine when it should re-initialize a counter.

Detailed Description Text (117):
Illustratively, the threshold for an interrupt modulator's time counter is determined, at least in part, by the amount of time that the polling software is expected to need in order to process an expected amount of traffic. The time counter threshold, and the frequency of polling, may also be affected by the speed of a host computer processor.

CLAIMS:

30. The interrupt modulator of claim 29, further comprising: a first packet threshold storage device configured to store a first initial packet count; a second packet threshold storage device configured to store a second initial packet count; a packet counter configured to hold a packet count, wherein said packet count is incrementable in response to transfer of packets; wherein said interrupt generator is further configured to generate an interrupt in response to said transfer of said packet if said packet count reaches a final packet count; and wherein said packet count is reset to one of said first initial packet count and said second initial packet count in response to said poll or said termination of processing of said interrupt.

L2: Entry 4 of 29 File: USPT Aug 13, 2002

DOCUMENT-IDENTIFIER: US 6434651 B1
TITLE: Method and apparatus for suppressing interrupts in a high-speed network environment

Abstract Text (1):
A network interface is polled by a process operating on a host computer system. Each time the network interface is polled, the process determines whether any packets have been received. If so, they are processed. Interrupts that would normally be issued by the network interface in response to the transfer of packets to the host system are suppressed or postponed during the polling mode of operation. If, however, a predetermined period of time has elapsed or a predetermined number of packets have been received since a previous poll or a previous interrupt, then an interrupt may be generated. The rate at which interrupts may be issued is modulated to decrease the interrupt-processing burden placed on the processor. A time counter may be used to track the passage of time and a packet counter may be used to track the number of packets transferred by the network interface. After each polling operation or processing of an interrupt by the host processor, the time and packet counters are reset to threshold values and thereafter begin decrementing toward a final time count and a final packet count, respectively. Thus, a packet transferred after one polling operation or interrupt does not cause the issuance of an interrupt to the host processor unless a time or packet counter reaches its final value (e.g. zero). The threshold time count and packet count may be adjusted to ensure that interrupts are generated often enough to avoid a negative impact on the processing of packets if, for example, the polling process is blocked.

Brief Summary Text (12):
In one embodiment of the invention, a system and method for polling a network interface are provided. In this embodiment an interrupt that would normally alert a host processor to the arrival of a network packet is suspended during a polling mode of operation. Each time the network interface is polled, any waiting packets are processed. However, if a threshold amount of time passes, or a threshold number of packets are received without being processed, interrupts may be enabled to ensure the packets are serviced. Thus, a polling mode of operation may be combined with interrupt modulation in one embodiment of the invention.

Brief Summary Text (15):
Each time the network interface is polled, and upon the completion of processing of an interrupt by a host processor, a time counter and/or a packet counter are initialized. Illustratively, the counters are set to programmable threshold values representing, respectively, a maximum period of time that is allowed to pass and a maximum number of packets that may be received before issuing another interrupt. After initialization, the counters begin decrementing toward final programmable values (e.g., zero). As long as polling continues in this embodiment, the counters will be repeatedly re-initialized and will therefore never expire and no interrupts will be generated. In an alternative embodiment, the counters are initialized to initial values and thereafter increment toward threshold values.

Detailed Description Text (22):
Illustratively, the time counter and packet counter are set to threshold values

after a host computer processes an interrupt. Suitable threshold values, which may be stored in programmable registers or other data structures, are twenty microseconds of time and seven packets for a moderate level of traffic. Thereafter, the time counter decrements (e.g., counts down toward zero) in response to a clock signal or other means of noting the passage of time, and the packet counter decrements in response to a packet transfer signal or other indicator that a packet was transferred to the host computer. For example, a packet transfer signal may be generated by a NIC module responsible for copying a packet to host memory. Until either the time counter or packet counter reaches zero or some other final value, which may be stored in a register or other programmable data storage unit, an interrupt will not be generated in response to the transfer of a packet. After a final value is reached, an interrupt may be generated for a packet transferred after the last interrupt,was processed by the host computer.

Detailed Description Text (25):
Interrupt modulation according to the present invention differs from that of the '758 patent in how the time and packet counters are reset. In particular, rather than being reset at the time the interrupt is issued, in a present embodiment of the invention the counters are not reset until after the host computer processes an interrupt. As described above, in one embodiment of the invention a second interrupt is not initiated by a network interface until a minimum period of time passes or a specified number of packets are received after a host processor completes servicing a first interrupt. One skilled in the art will recognize that the present scheme provides additional time separation between successive interrupts and thus further decreases the amount of host processor time expended in servicing interrupts from the network interface. Under the '758 patent, if a host processor requires nearly a full threshold of time (e.g., twenty microseconds) to finish its processing of one interrupt, interrupts may be enabled and an interrupt issued shortly after the first one is serviced.

Detailed Description Text (26):
FIG. 1 is a diagram of an interrupt modulator for a NIC according to one embodiment of the invention. In this embodiment time and packet counters are set to their threshold values (e.g. twenty microseconds and seven packets, respectively) after an interrupt is processed by a host computer and then decremented in accordance with the passage of time and the transfer of packets. A counter expires when it reaches a final value (e.g., zero), at which time another interrupt may be generated. Although threshold values of approximately twenty microseconds and approximately seven packets are employed in embodiments of the invention discussed below, in alternative embodiments a wide range of thresholds will be suitable. In particular, one network environment in which an embodiment of the invention may be practiced (e.g., such as the Internet) employs Ethernet, IP and TCP protocols, respectively, at layers two, three and four of an associated protocol stack. In this environment a range of twenty to fifty microseconds may be appropriate for a time counter and a range of four to ten packets may be appropriate for a packet counter.

Detailed Description Text (30):
The time threshold registers and packet threshold registers depicted in FIG. 1 are implemented in programmable memory (e.g., register, RAM, flash memory). Thus, they may be set or modified by software (e.g., a device driver) operating on a host computer. As depicted in FIG. 1, multiple threshold registers may be coupled to a time and/or packet counter in order to allow the counters to be reset to different values depending on the level of network traffic. In particular, different threshold values may be desirable depending on the level of traffic received at the NIC or processed by a host processor. The time and packet counters may therefore be re-initialized to different thresholds as the traffic fluctuates. An interrupt modulator is not limited to a particular number of threshold registers and may include any number of them greater than or equal to one. In another alternative embodiment, threshold time and/or packet counts are stored in read-only memory.

Detailed Description Text (36):
When initialized, time counter 102 is set to the threshold value stored in time threshold register 104a (e.g., or an alternate time threshold register such as register 104b). As the time counter is decremented in response to clock signal 110, it is compared to the value in final time register 116 by comparator 118. If time counter 102 is decremented to or beyond the value in final time register 116 (e.g., indicating the passage of at least twenty microseconds of time), then time expired signal 120 is activated. The value stored in time threshold register 104a may be modified (e.g., increased) as the rate of receipt of network traffic increases. Or, as shown in FIG. 1, a different threshold value may be loaded from a different time threshold register depending upon the level of traffic (e.g., number of packets processed in an interrupt, number of packets received in a given period of time). The determination of which time threshold register's value to use for initializing time counter 102 may be made on the basis of the amount of traffic being transferred to the host computer. The value in final time register 116 may also be altered.

Detailed Description Text (37):
Similarly, when packet counter 106 is initialized, it is set to the threshold value stored in packet threshold register 108a (or an alternate packet threshold register such as register 108b). As the packet counter is decremented in response to packet transfer signal 112, it is compared to the value in final packet register 122 by comparator 124. If packet counter 106 is decremented to or beyond the value in final packet register 122, then packets exceeded signal 126 is activated. The value stored in packet threshold register 108a may be modified (e.g., increased) as the rate of receipt of network traffic increases. Or, as shown in FIG. 1, a different threshold value may be loaded from a different packet threshold register depending upon the level of traffic (e.g., number of packets processed in an interrupt, number of packets received in a given period of time). The value in final packet register 122 may also be altered.

Detailed Description Text (51):
In state 304, interrupts associated with the transfer of packets from a network interface are temporarily disabled. Certain steps toward the issuance of an interrupt may still be performed in response to the transfer of a packet, such as the setting of an indicator in a status register. As described below, however, no packet transfer interrupt will be transmitted to a host processor until either the time or packet counter reaches its threshold value.

Detailed Description Text (52):
In state 306, one or both of the time and packet counters decrement (e.g., increment negatively, or toward zero). In particular, the time counter will generally continuously decrement, from the time it is initialized, until it is re-initialized. It may, however, be halted or suspended (e.g., cease decrementing) when interrupts are enabled. After an interrupt is processed the time counter is re-initialized to a threshold value and once again begins decrementing. A packet counter, however, decrements in response to an event that is less certain than the passage of time--the transfer of a packet from the network interface. Illustratively, for each packet transferred by the network interface to a host computer the packet counter count decreases by one.

Detailed Description Text (60):
When packets are transferred from a network interface at a very high rate, a packet counter's threshold may be quickly surmounted each time it is reset. Each time the threshold is exceeded another interrupt would then be generated to a host processor. Of course the packet counter threshold may be set to a relatively high value in response (e.g., by loading a threshold value from a different packet threshold register), but if the transfer rate remains very high then each time the processor receives an interrupt it may expend an inordinate amount of time

processing a large number of packets. Some communication protocols may not be able to adequately function in such an environment. Or, if the rate of packet transfer to the host computer decreases precipitously after the packet counter threshold is increased, then the next interrupt, and the subsequent processing of one or more packets, may be delayed.

Detailed Description Text (61):
Even the use of a time counter may not alleviate the problems associated with heavy traffic levels. In particular, if the time counter has a relatively high time threshold, the number of packets transferred by a network interface before the threshold is exceeded may again require the processor to spend a significant period of time processing the packets when an interrupt is generated. If the time threshold is set to a low value (e.g., by using a different time threshold register), then the processor may be over-burdened with interrupts--similar to the result of employing a low packet counter threshold.

Detailed Description Text (80):
From decrement counter state 410, polling state 406 may be entered because a polling timer expires or reaches its threshold, as represented by transition 416. A polling timer's threshold is preferably lower than the interrupt modulator's time threshold to reflect the preference for a polling operation over the generation of an interrupt. If, however, the time counter expires or reaches its final value before a polling operation can be completed (after which the time counter is re-initialized), transition 418 illustrates the entry into interrupt enabled state 404 from decrement counter state 410. Transition 418 may also be initiated by the receipt of a threshold number of packets. From interrupt enabled state 404, initialize counter state 408 is entered after an interrupt is serviced by a host processor, as indicated by transition 420.

Detailed Description Text (103):
Illustratively, the status register is accessed through an alias register (e.g., alias address) during heavy levels of network traffic (e.g., when a polling mode of operation is active). Otherwise, the status register may be accessed directly. In addition, it was described above that different threshold values for time and/or packet counters in an interrupt modulator may be desirable for different levels of traffic. Therefore, in an embodiment of the invention in which an interrupt modulator stores multiple time and/or packet threshold values (e.g., in separate threshold registers), the threshold value that is loaded into a counter may be determined by how the status register is accessed.

Detailed Description Text (107):
In state 704, it is determined whether the level of traffic or rate of interrupts or some other traffic measure indicates that it would be more efficient to implement a polling mode of operation. For example, it may be determined that the number of packets processed during an interrupt exceeds a programmable threshold (e.g., fifty). Other criteria for making this determination include the rate at which interrupts are issued by the network interface for packets transferred to the host (e.g., approximately 10,000 per second) or the level of processor utilization (e.g., approximately ninety percent). These thresholds may be measured on a one-time or instantaneous basis, or may be averaged or otherwise combined over a period of time or a number of interrupts. The illustrated procedure returns to state 702 if an applicable threshold is not met.

Detailed Description Text (109):
In state 708 thresholds for one or more interrupt modulator counters (e.g., a time and/or packet counter) are set or increased from lower values used during interrupt modulation. The counters are activated and thereafter decrement or increment as described previously. By increasing the thresholds, any interrupts that may be issued by the interrupt modulator are issued with less frequency, if at all. Delaying interrupts allows the polling module to perform its polling and packet

processing and allows a host processor to avoid the overhead involved in servicing an interrupt.

Detailed Description Text (110):
In this embodiment an interrupt modulator may store multiple threshold values in multiple registers for a packet and/or a time counter in order to facilitate the modification of counter thresholds. With multiple threshold registers, the counters may be easily re-initialized or switched to use different thresholds. In particular, when operating in a moderate level of network traffic (e.g., without using an alias register) a packet or time counter may be re-initialized to a first threshold every time a status register is cleared (e.g., because of an interrupt). However, when operating in a heavier traffic environment characterized by the use of polling and, possibly, an alias register, a counter may be re-initialized to a second threshold when the alias register is read by the polling software.

Detailed Description Text (113):
State 710 demonstrates that if an interrupt modulator reaches a threshold value, the network interface will not continue waiting for a polling operation. In one alternative embodiment a polling timer may be examined more frequently than a packet or time counter. In other words, states 710-712 and 714-716 may be reversed, such that the interrupt modulator's counters are only examined when it is determined that it is not time for a polling operation.

Detailed Description Text (116):
During polling in a heavy traffic environment the software may have many packets to process each time it examines a network interface's status register or alias register. In order to prevent an interrupt modulator's counter from reaching its threshold during the processing of the packets, the polling software may temporarily pause, suspend or interrupt operation in order to re-initialize the counter(s). For example, if an interrupt modulator's time counter is set to one millisecond, the polling software will attempt to ensure that it re-initializes the time counter less than one millisecond after it was last re-initialized. The polling software may examine the time counter, its own counter, or another counter or clock signal in order to determine when it should re-initialize a counter.

Detailed Description Text (117):
Illustratively, the threshold for an interrupt modulator's time counter is determined, at least in part, by the amount of time that the polling software is expected to need in order to process an expected amount of traffic. The time counter threshold, and the frequency of polling, may also be affected by the speed of a host computer processor.

CLAIMS:

30. The interrupt modulator of claim 29, further comprising: a first packet threshold storage device configured to store a first initial packet count; a second packet threshold storage device configured to store a second initial packet count; a packet counter configured to hold a packet count, wherein said packet count is incrementable in response to transfer of packets; wherein said interrupt generator is further configured to generate an interrupt in response to said transfer of said packet if said packet count reaches a final packet count; and wherein said packet count is reset to one of said first initial packet count and said second initial packet count in response to said poll or said termination of processing of said interrupt.

First Hit Fwd Refs

Generate Collection Print

L2: Entry 4 of 29 File: USPT Aug 13, 2002

US-PAT-NO: 6434651
DOCUMENT-IDENTIFIER: US 6434651 B1

TITLE: Method and apparatus for suppressing interrupts in a high-speed network environment

DATE-ISSUED: August 13, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gentry, Jr.; Denton E.	Fremont	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Sun Microsystems, Inc.	Palo Alto	CA			02

APPL-NO: 09/ 260367 [PALM]
DATE FILED: March 1, 1999

INT-CL: [07] G06 F 9/48

US-CL-ISSUED: 710/260; 709/235
US-CL-CURRENT: 710/260; 709/235

FIELD-OF-SEARCH: 710/260, 709/235, 709/250

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5319752	June 1994	Petersen et al.	
5485584	January 1996	Hausman et al.	
5659758	August 1997	Gentry et al.	
5797037	August 1998	Ecclesine	
6065073	May 2000	Booth	370/908
6105102	August 2000	Williams et al.	710/261
6256660	July 2001	Govindaraju et al.	709/200

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0752799	January 1997	EP	
0852357	July 1998	EP	
2 282 474	April 1995	GB	

OTHER PUBLICATIONS

Pending U.S. patent application Ser. No. 09/259,445, entitled "Method and Apparatus for Distributing Network Processing on a Multiprocessor Computer," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3481-JTF).
Pending U.S. patent application Ser. No. 09/259,736, entitled "Method and Apparatus for Modulating Interrupts in a Network Interface," by Denton Gentry et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3483-JTF).
Pending U.S. patent application Ser. No. 09/259,765, entitled "A High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3485-JTF).
Pending U.S. patent application Ser. No. 09/260,618, entitled "Method and Apparatus for Classifying Network Traffic in a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3486-JTF).
Pending U.S. patent application Ser. No. 09/259,932, entitled "Method and Apparatus for Managing a Network Flow in a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3487-JTF).
Pending U.S. patent application Ser. No. 09/260,324, entitled "Method and Apparatus for Dynamic Packet Batching with a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3488-JTF).
Pending U.S. patent application Ser. No. 09/258,952, entitled "Method and Apparatus for Early Random Discard of Packets," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3490-JTF).
Pending U.S. patent application Ser. No. 09/260,333, entitled "Method and Apparatus for Data Re-Assembly with a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3507-JTF).
Pending U.S. patent application Ser. No. 09/258,955, entitled "Dynamic Parsing in a High Performance Network Interface," by Denton Gentry, filed Mar. 1, 1999 (Attorney Docket SUN-P3715-JTF).
Pending U.S. patent application Ser. No. 09/259,936, entitled "Method and Apparatus for Indicating an Interrupt in a Network Interface," by Denton Gentry et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3814-JTF).

ART-UNIT: 2181

PRIMARY-EXAMINER: Auve; Glenn A.

ATTY-AGENT-FIRM: Park, Vaughan & Fleming LLP

ABSTRACT:

A network interface is polled by a process operating on a host computer system. Each time the network interface is polled, the process determines whether any packets have been received. If so, they are processed. Interrupts that would normally be issued by the network interface in response to the transfer of packets to the host system are suppressed or postponed during the polling mode of operation. If, however, a predetermined period of time has elapsed or a predetermined number of packets have been received since a previous poll or a previous interrupt, then an interrupt may be generated. The rate at which

interrupts may be issued is modulated to decrease the interrupt-processing burden placed on the processor. A time counter may be used to track the passage of time and a packet counter may be used to track the number of packets transferred by the network interface. After each polling operation or processing of an interrupt by the host processor, the time and packet counters are reset to threshold values and thereafter begin decrementing toward a final time count and a final packet count, respectively. Thus, a packet transferred after one polling operation or interrupt does not cause the issuance of an interrupt to the host processor unless a time or packet counter reaches its final value (e.g. zero). The threshold time count and packet count may be adjusted to ensure that interrupts are generated often enough to avoid a negative impact on the processing of packets if, for example, the polling process is blocked.

53 Claims, 7 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L2: Entry 7 of 29 File: USPT Aug 14, 2001

DOCUMENT-IDENTIFIER: US 6275502 B1
**** See image for Certificate of Correction ****
TITLE: Advanced priority statistical multiplexer

Detailed Description Text (220):
The resulting data transmissions can be characterized by various indices such as high priority packet time (HPPT) which is a measure of the time high priority packet module 2001 needs to construct a high priority packet. The high priority packet time is commensurate with the sampling rate of the high priority input and is dictated by the nature of the high priority signal, level of compression of that signal, and requisite bandwidth. Each high priority packet is loaded into shared memory 2002 by high priority packet module 2001 for later transmission by aggregate module 2005. Aggregate module 2005 transceives the packets at the same rate as high priority packet module 2001 so each packet will be transferred within one high priority packet time, ensuring that the high priority data is timely. If the high priority packet module 2001 is the voice/fax card 308, then the high priority packet time is dependent on the speech compression algorithm selected. For example, the earlier section entitled "Speech Compression Algorithm" described a 20 ms speech sample time. In this case the high priority packet time would be 20 ms, since voice packets are generated and must be processed every 20 ms. The high priority packet time multiplied by the overall baud transmission rate of the link sets the maximum bandwidth (in bytes) which may be allocated to high priority packet transmission, known as HPPT.sub.n. Another index is the interrupt boundary byte count (IBBC), which is the excess overhead of the communications channel assuming the maximum number of high priority packet bytes were continually transmitted. The calculation of the interrupt boundary byte count is described below by the following pseudocode procedures:

Detailed Description Text (246):
Using this algorithm, aggregate module 2005 polls for high priority data at the beginning of each transmission of a frame and in IBBC byte intervals measured from the transmission of the last high priority data byte. In one embodiment of the present invention the low priority packet module 2003 transfers packetized data to common memory 2004 upon three conditions: (1) reaching a predetermined maximum low priority packet packet byte count; (2) when a flash timer signals the transfer prior to filling the packet up to the packet byte count; or (3) if a high priority header occurs on the IBBC+1th byte in the hybrid stream. Therefore, X may be less than IBBC, since the low priority packet byte count is less than IBBC bytes in cases (1) and (2).

Detailed Description Text (273):
FIG. 20B shows the output from one embodiment of a two priority level advanced priority statistical multiplexer. Data segment 2020 is an enlargement of one segment of duration equal to one high priority packet time taken from an output data stream 2010. The number of bytes which can be transmitted in one high priority packet time is HPPT.sub.n 2012. In order to graphically illustrate the interrupt boundary byte count, the high priority portion of the segment 2020 demonstrates the maximum number of high priority data bytes which can be transmitted in one high priority packet time, HPPT.sub.sum 2022. The difference between HPPT.sub.n 2012 and HPPT.sub.sum 2022 is the interrupt boundary byte count 2024. However, in ordinary

transmissions the high priority portion of a segment may have anywhere from zero to HPPT.sub.sum bytes of high priority data.

CLAIMS:

37. A method for multiplexing high priority data, intermediate priority data, and low priority data for transmission across a communications link, the link having a maximum bandwidth with a byte transfer period equal to an amount of time needed to transfer one byte of data over the communications link, the method comprising the steps of:

a. determining a high priority packet time;

b. determining a maximum number of bytes transferred in one high priority packet time;

c. determining a maximum number of high priority data bytes transferred in one high priority packet time;

d. subtracting the maximum number of high priority bytes from the maximum number of bytes to obtain an interrupt boundary byte count;

e. multiplexing high priority data, intermediate priority data, and low priority data, wherein the step of multiplexing includes the steps of:

First Hit Fwd Refs

☐ Generate Collection Print

L2: Entry 7 of 29 File: USPT Aug 14, 2001

US-PAT-NO: 6275502
DOCUMENT-IDENTIFIER: US 6275502 B1
**** See image for Certificate of Correction ****

TITLE: Advanced priority statistical multiplexer

DATE-ISSUED: August 14, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Arimilli; Harinarayana	Coon Rapids	MN		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Multi-Tech Systems, Inc.	Mounds View	MN			02

APPL-NO: 08/ 885534 [PALM]
DATE FILED: June 30, 1997

PARENT-CASE:
This application is a continuation of U.S. patent application Ser. No. 08/333,365, filed Nov. 2, 1994 now issued as U.S. Pat. No. 5,757,801. This is a continuation in part of copending U.S. patent application Ser. No. 08/229,958 Now U.S. Pat. No. 5,682,386 which is filed Apr. 19, 1994 now issued as U.S. Pat. No. 5,757,801 entitled "Data/Voice/Fax Compression Multiplexer" which is hereby incorporated by reference.

INT-CL: [07] H04 J 3/16, H04 J 3/22

US-CL-ISSUED: 370/468; 370/477, 370/537, 370/471
US-CL-CURRENT: 370/468; 370/471, 370/477, 370/537

FIELD-OF-SEARCH: 370/229, 370/230, 370/232, 370/235, 370/391, 370/412, 370/464, 370/465, 370/468, 370/477, 370/493, 370/435, 370/437, 370/538, 370/539, 370/540, 370/541, 370/542, 370/543

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	Re34034	August 1992	O'Sullivan	379/59

3304372	February 1967	Filipowsky et al.	179/2
3781818	December 1973	Pardoe et al.	370/538
3789165	January 1974	Campanella	
3904830	September 1975	Every, Sr. et al.	
3973081	August 1976	Hutchins	
3997732	December 1976	Every, Sr. et al.	
4100377	July 1978	Flanagan	
4107471	August 1978	Reed	
4205202	May 1980	Kahn	
4284850	August 1981	Clingenpeel	
4354273	October 1982	Araseki et al.	
4377860	March 1983	Godbole	
4403322	September 1983	Kato et al.	
4425661	January 1984	Moses et al.	
4445213	April 1984	Baugh et al.	
4446555	May 1984	Devault et al.	370/352
4476559	October 1984	Brolin et al.	
4479195	October 1984	Herr et al.	
4479213	October 1984	Galand et al.	
4495620	January 1985	Steele et al.	
4500987	February 1985	Hasegawa	
4524244	June 1985	Faggin et al.	
4534024	August 1985	Maxemchuk et al.	
4546212	October 1985	Crowder, Sr.	
4549291	October 1985	Renoulin et al.	370/452
4578537	March 1986	Faggin et al.	
4587651	May 1986	Nelson et al.	
4593389	June 1986	Wurzburg et al.	
4598397	July 1986	Nelson et al.	
4609788	September 1986	Miller et al.	
4610022	September 1986	Kitayama et al.	
4629829	December 1986	Puhl et al.	
4652703	March 1987	Lu et al.	
4660218	April 1987	Hashimoto	
4670874	June 1987	Sato et al.	
4697281	September 1987	O'Sullivan	
4700341	October 1987	Huang	
4707831	November 1987	Weir et al.	

4718082	January 1988	Parker et al.	
4740963	April 1988	Eckley	
4750169	June 1988	Carse et al.	
4751510	June 1988	De Saint Michel et al.	
4751736	June 1988	Gupta et al.	
4757527	July 1988	Beniston et al.	
4764955	August 1988	Galand et al.	
4794595	December 1988	Ohyama	
4807250	February 1989	Tanaka	
4809271	February 1989	Kondo et al.	
4813040	March 1989	Futato	
4827085	May 1989	Yaniv et al.	
4835765	May 1989	Bergmans et al.	
4839802	June 1989	Wonak et al.	
4845746	July 1989	Li	
4847900	July 1989	Wakim	
4862449	August 1989	Hoefkens et al.	
4864559	September 1989	Perlman	
4866732	September 1989	Carey et al.	
4873715	October 1989	Shibata	
4884264	November 1989	Servel et al.	370/422
4887265	December 1989	Felix	
4890282	December 1989	Lambert et al.	
4901333	February 1990	Hodgkiss	
4905282	February 1990	McGlynn et al.	
4912756	March 1990	Hop	
4912758	March 1990	Arbel	
4914650	April 1990	Sriram	
4926416	May 1990	Weik	370/354
4926448	May 1990	Kraul et al.	
4932048	June 1990	Kenmochi et al.	
4935954	June 1990	Thompson et al.	
4942569	July 1990	Maeno	
4953210	August 1990	McGlynn et al.	
4965789	October 1990	Bottau et al.	
4972457	November 1990	O'Sullivan	
4972462	November 1990	Shibata	

☐				
☐	4972483	November 1990	Carey	
☐	4977591	December 1990	Chen et al.	
☐	4991169	February 1991	Davis et al.	
☐	4993022	February 1991	Kondo et al.	370/462
☐	4995059	February 1991	Ishikawa	
☐	4998241	March 1991	Brox et al.	
☐	5001710	March 1991	Gawrys et al.	
☐	5001745	March 1991	Pollock	
☐	5005183	April 1991	Carey et al.	
☐	5008901	April 1991	Wallach et al.	
☐	5008926	April 1991	Misholi	
☐	5014232	May 1991	Andre	
☐	5020058	May 1991	Holden et al.	
☐	5025443	June 1991	Gupta	
☐	5036513	July 1991	Greenblatt	
☐	5044010	August 1991	Frenkiel et al.	
☐	5046188	September 1991	Molnar	
☐	5051720	September 1991	Kittirutsunetorn	
☐	5062133	October 1991	Melrose	
☐	5065395	November 1991	Shenoi et al.	
☐	5065425	November 1991	Lecomte et al.	
☐	5081647	January 1992	Bremer	
☐	5083310	January 1992	Drory	
☐	5086471	February 1992	Tanaka et al.	
☐	5099472	March 1992	Townsend et al.	
☐	5107519	April 1992	Ishikawa	
☐	5115429	May 1992	Hluchyj et al.	
☐	5121385	June 1992	Tominaga et al.	
☐	5127001	June 1992	Steagall et al.	
☐	5127041	June 1992	O'Sullivan	
☐	5132966	July 1992	Hayano et al.	370/233
☐	5136586	August 1992	Greenblatt	
☐	5138662	August 1992	Amano et al.	
☐	5146470	September 1992	Fujii et al.	
☐	5150410	September 1992	Bertrand	
☐	5151937	September 1992	Chujo et al.	
	5153897	October 1992	Sumiyoshi et al.	

☐				
☐	5162812	November 1992	Aman et al.	
☐	5164982	November 1992	Davis	
☐	5177734	January 1993	Cummiskey et al.	
☐	5182762	January 1993	Shirai et al.	
☐	5187591	February 1993	Guy et al.	
☐	5187692	February 1993	Haneda et al.	
☐	5193110	March 1993	Jones et al.	
☐	5195130	March 1993	Weiss et al.	
☐	5208812	May 1993	Dudek et al.	
☐	5208850	May 1993	Kino	
☐	5214656	May 1993	Chung et al.	
☐	5227876	July 1993	Cucchi et al.	370/235
☐	5228026	July 1993	Albrow et al.	
☐	5233660	August 1993	Chen	
☐	5235595	August 1993	O'Dowd	
☐	5249218	September 1993	Sainton	
☐	5258983	November 1993	Lane et al.	
☐	5261027	November 1993	Taniguchi et al.	
☐	5263019	November 1993	Chu	
☐	5272695	December 1993	Makino et al.	
☐	5276703	January 1994	Budin et al.	
☐	5278900	January 1994	Van Gerwen et al.	
☐	5280479	January 1994	Mary	370/462
☐	5282197	January 1994	Kreitzer	
☐	5283638	February 1994	Engberg et al.	
☐	5283819	February 1994	Glick et al.	
☐	5289539	February 1994	Maruyama	
☐	5295136	March 1994	Ashley et al.	
☐	5307413	April 1994	Denzer	
☐	5309562	May 1994	Li	
☐	5313498	May 1994	Sano	
☐	5317604	May 1994	Osterweil	
☐	5319682	June 1994	Clark	
☐	5327520	July 1994	Chen	
☐	5329472	July 1994	Sugiyama	
☐	5341374	August 1994	Lewen et al.	
	5343473	August 1994	Cidon et al.	

☐				
☐	5343521	August 1994	Jullien et al.	
☐	5355365	October 1994	Bhat et al.	
☐	5365577	November 1994	Davis et al.	
☐	5371853	December 1994	Kao et al.	
☐	5379292	January 1995	Kurata et al.	370/216
☐	5384780	January 1995	Lomp et al.	
☐	5390239	February 1995	Morris et al.	
☐	5406557	April 1995	Baudoin	
☐	5414796	May 1995	Jacobs	
☐	5416776	May 1995	Panzarella et al.	
☐	5438614	August 1995	Rozman et al.	
☐	5444770	August 1995	Davis et al.	
☐	5463616	October 1995	Kruse et al.	370/24
☐	5472351	December 1995	Greco et al.	
☐	5473676	December 1995	Frick et al.	
☐	5479407	December 1995	Ko et al.	
☐	5490060	February 1996	Malec et al.	
☐	5493609	February 1996	Winseck, Jr. et al.	

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
3504064	August 1986	DE	
3630469	March 1988	DE	
3409532	April 1989	DE	
0 429 054 A3	May 1991	EP	
488865A2	June 1992	EP	
0 650 286 A2	April 1994	EP	
2210237A	January 1989	GB	
63-054052	August 1988	JP	
193489	July 1990	JP	
257748	October 1990	JP	
WO 92/06550	April 1992	WO	
WO 93/22869	November 1993	WO	
WO 94/26056	November 1994	WO	

OTHER PUBLICATIONS

European Search Report for Application No. EP 93403164 completed on Sep. 21, 1995 by Examiner Lambley; 4 pages.
S. Casale et al., Statistical Voice/High-Speed Data Multiplexing on a 64 KBIT/S Channel, IEEE, pp. 459-464, dated 1991.

T. Komiya et al, "An Approach to the Multifunction Graphic Terminal for the ISDN Environment", IEEE, pp. 32-36, dated 1988.
D. Gulick et al., "Interface for the ISDN to Your PC with A Voice/Data Board", Electronic Design, pp. 85-88, dated Dec. 10, 1987.
S. Sasaki et al., "Variable Rate Voice Coding System", IEEE, pp. 364-367, dated 1992.
"Video Calls Use Basic Phone Lines", Mitch Radcliffe, MacWeek, (Aug. 3, 1992).
"Radish System Lets Phone Users Send Voice, Data Simultaneously", PC Week, 9, 19, pp. 53, (May 11, 1992).
"Don't Just Tell Them, Show Them!", Glenn A. Pierce, Jr., Automation, (Aug. 1990).
"Mitsubishi Still Color Picture TV Phone", Techno Japan, 23, 6, (Jun. 1990).
"The Photophone", (Product Brochure) GTE (Feb. 15, 1990).
"Wrist TVs Still Fiction, but Police Videophone Take Hold", Ray Smith, TE&M, (Dec. 15, 1987).
"Audiographic Terminal", M. Laube, Electrical Communication, 60, 1, (1986).
"Comparison of Coding Schemes for Telewriting Systems", Tominaga et al., ICCC, (1986).
"Simultaneous Transmission of Voice and Handwriting Signals: Sketchphone System", Kishimoto et al., IEEE, (1981).
"Telewriting Terminal Equipment" (Recommendation T.150) CCITT, (1988).
"A Family of 2-Wire, Duplex Modems Operating at Data Signalling Rates . . . ", Facsimile Recommendation V.32 CCITT, (1988).
"*** PICFON Card Brochure", Specom Technologies Corp., (Published Prior to Applicant's Invention).
Pen Telephone Brochure, Shimadzu, (Published Prior to Applicant's Invention).
"Telewriter Product Description", Optel Communications, Inc., (Published Prior to Applicant's Invention).
"Videowriter '91 Product Description", Optel Communications, Inc., (1991).
Copy of PCT Search Report for Application Serial No. PCT/US 96/11313 completed on Nov. 7, 1996, by C. Mikkelsen, 4 pages.
IBM Technical Disclosure Bulletin, Speech Data Adaptive Multiplexer, 27, No. 2, pp. 969, dated Jul. 1994.
Copy of PCT Search Report dated Apr. 25, 1996 by Areste Canosa for Application No. PCT/US95/05034 (8 pages).
Copy of PCT Search Report dated Apr. 10, 1996 by J. Lange for Application No. PCT/US95/14826 (7 pages).
Copy of PCT Search Report dated Mar. 28, 1996 by M. Vandevenne for Application No. PCT/US95/14829 (8 pages).
Copy of PCT Search Report dated May 24, 1996 by C. Canosa Areste for Application No. PCT/US 95/14253 (6 pages).
Copy of European Search Report dated Apr. 18, 1996 by S. Lambley for Application No. EP 93403164 (5 pages).
Canadian Application No. 2,104,701, Computer-Based Multifunction Personal Communications System, pp. 1-105, and 52 sheets of drawings, dated Jul. 9, 1994.
AT&T Microelectronics, "High Speed Data Pump Chip Sets," published in December 1991.
AT&T Microelectronics, "WE DSP16C Digital Signal Processor/CODEC Preliminary Data Sheet," 32 pages, published in May, 1991.
AT&T Microelectronics, "T7540 Digital Telephone CODEC Data Sheet Addendum," pp. 1-4, published in Jul., 1991.
AT&T Microelectronics, "T7540 Digital Telephone CODEC Preliminary Data Sheet," pp. 1-64, published in Jan., 1991.
Zilog Intelligent Peripheral Controllers, "Z84C01 Z80 CPU with Clock Generator/Controller," pp. 43-73, published in 1991.
Zilog Intelligent Peripheral Controllers, "Z84C90 CMOS Z80 KIO Serial/Parallel/counter/timer," pp.205-224, published in 1991.
U.S. West Caller ID publication, received Jul. 18, 1994, one page.
J.D. Mills, et al., "A Data and Boice System for the General Service Telephone Network," IECON, pp. 1143-1148, 1987.
Copy of European Search Report (Application No. EP 94304742), completed Jun. 8,

1995 by Examiner Mikkelsen.
"TechTips--A Periodic Round-up of Technical Applications, Notes and Information on MultiTech's Data Communications Products" by MultiTech Systems, vol. 2, No. 2, May 1992.
"MultiX25--X.25 PAD, The New MultiX25 PAD 8 Port X.25 Packet Assembler/Disassembler for Public and Private Data Networks," y MultiTech Systems.
Y. Akaiwa et al., "An Integrated Voice and Data Radio Access System," 1992, pp.255-258, IEEE.
CCITT V.42, "Error-Correcting Procedures for DCES Using Asynchronous-to-Synchronous Conversion", vol. VIII, pp. 296-370, dated 1988.

ART-UNIT: 272

PRIMARY-EXAMINER: Patel; Ajit

ASSISTANT-EXAMINER: Phunkulh; Bob A.

ATTY-AGENT-FIRM: Schwegman, Lundberg, Woessner & Kluth, P.A.

ABSTRACT:

A data multiplexing network is described which multiplexes a plurality of asynchronous data channels with an asynchronous data stream representing compressed voice signals and/or facsimile signals onto a single synchronous data packet stream. The single synchronous data packet stream is then transmitted by a high speed statistical multiplexer over a composite link to a second site using a modified high-level synchronous data link control protocol with an overlay of an advanced priority statistical multiplexing algorithm. The asynchronous data channels and the compressed voice channel and/or facsimile signals are demultiplexed and reconstructed for sending to other asynchronous computer terminals and to a standard telephone or facsimile analog port at the second site, respectively. PBX trunk interfaces are also provided to allow PBX's to share the composite link between sites. Communication between the first site by voice or facsimile and the second site is transparent to the users. The multiplexer efficiently allocated the bandwidth of the composite link by detecting silence periods in the voice signals and suppressing the sending of the voice information to preserve bandwidth. An advanced priority statistical multiplexer is also described which dynamically allocates composite link bandwidth to both time-sensitive and non-time-sensitive data to maximize data throughout efficiency and quality while simultaneously reducing multiplexer processing overhead.

41 Claims, 65 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L2: Entry 8 of 29 File: USPT Apr 10, 2001

DOCUMENT-IDENTIFIER: US 6216182 B1
TITLE: Method and apparatus for serving data with adaptable interrupts

Brief Summary Text (6):
Other existing methods use a counter or timer to limit the number of interrupts but can have latency problems especially for video or voice data. An interrupt is generated every N packet and M clock ticks from the time the last packet was enqueued. This method attempts to minimize the number of interrupt by generating an interrupt after N packets are enqueued (N is programmable). To minimize latence in case N is large, an interrupt is generated based on a timer. The timer is triggered at the end of a packet. If the timer crosses a M threshold (programmable) without detecting the end of another packet, then an interrupt is generated. Using this scheme requires the programming of two parameters: N and M. It is difficult for the host to determine the optimum value for N and M for different load conditions and variations of the host and card.

Detailed Description Text (13):
In all the schemes described above, there is an additional condition that can generate an interrupt. An interrupt is generated if the number of buffers 18 pending in the queue 16 reaches a high threshold. This is needed to prevent overflowing the queue 16 in the case of a very large packet (the queue 16 is almost full, but the end of the packet is not yet received).

First Hit Fwd Refs

Generate Collection Print

L2: Entry 11 of 29 File: USPT Apr 4, 2000

DOCUMENT-IDENTIFIER: US 6046998 A
TITLE: ATM reference traffic system

Detailed Description Text (8):
A typical personal computer is provided with an operating system loaded from the hard drive when the computer boots. In the present invention, the preferred operating system is the Linux operating system. As with most operating systems, Linux schedules events using timers placed in a timer queue made up of a doubly linked list of timers. Each timer is characterized by the time at which it should expire, the function that should be called when the timer expires, and the data that should be passed to the called function. Thus, whenever the expiration time of the timer is less than the present time, the associated timer function is called and executed. With Linux, as with many other operating systems the timers have a maximum resolution of 10 milliseconds (msec.). This resolution is provided by timer chip 24, which usually generates an interrupt every 10 milliseconds. This is the shortest resolution with which events can be scheduled in the Linux kernel.

Detailed Description Text (15):
FIG. 6 illustrates RELOAD.sub.-- TIMER subroutine 600 which starts at step 602 called in response to step 502. Step 602 determines the time to the next scheduled event and also determines the time elapsed since the last update to the variable JIFFIES U. Step 604 then asks whether the difference between these values is less than a specified threshold value called TIMER.sub.-- DELTA which is equal to the time for processing an interrupt routine. That is, if the time to the next event is close to or less than the threshold, the answer in step 604 is yes and step 606 loads the value TIMER DELTA into timer chip 24. If the time to the next event is greater than TIMER.sub.-- DELTA, then the answer in step 604 is no and step 608 loads the time to the next event into timer chip 24. This is determined by subtracting the time elapsed since the last update to JIFFIES.sub.-- U from the scheduled time to the next event. Subroutine 600 then stops.

First Hit Fwd Refs

Generate Collection Print

L2: Entry 11 of 29 File: USPT Apr 4, 2000

DOCUMENT-IDENTIFIER: US 6046998 A
TITLE: ATM reference traffic system

Detailed Description Text (8):
A typical personal computer is provided with an operating system loaded from the hard drive when the computer boots. In the present invention, the preferred operating system is the Linux operating system. As with most operating systems, Linux schedules events using timers placed in a timer queue made up of a doubly linked list of timers. Each timer is characterized by the time at which it should expire, the function that should be called when the timer expires, and the data that should be passed to the called function. Thus, whenever the expiration time of the timer is less than the present time, the associated timer function is called and executed. With Linux, as with many other operating systems the timers have a maximum resolution of 10 milliseconds (msec.). This resolution is provided by timer chip 24, which usually generates an interrupt every 10 milliseconds. This is the shortest resolution with which events can be scheduled in the Linux kernel.

Detailed Description Text (15):
FIG. 6 illustrates RELOAD.sub.-- TIMER subroutine 600 which starts at step 602 called in response to step 502. Step 602 determines the time to the next scheduled event and also determines the time elapsed since the last update to the variable JIFFIES U. Step 604 then asks whether the difference between these values is less than a specified threshold value called TIMER.sub.-- DELTA which is equal to the time for processing an interrupt routine. That is, if the time to the next event is close to or less than the threshold, the answer in step 604 is yes and step 606 loads the value TIMER DELTA into timer chip 24. If the time to the next event is greater than TIMER.sub.-- DELTA, then the answer in step 604 is no and step 608 loads the time to the next event into timer chip 24. This is determined by subtracting the time elapsed since the last update to JIFFIES.sub.-- U from the scheduled time to the next event. Subroutine 600 then stops.

L2: Entry 19 of 29 File: USPT May 27, 1997

DOCUMENT-IDENTIFIER: US 5634015 A
TITLE: Generic high bandwidth adapter providing data communications between diverse communication networks and computer system

Detailed Description Text (626):
4. If the inbound queue timer expires or the inbound queue packet count exceeds the threshold limit, the GAM will generate an interrupt to the P.

Detailed Description Text (629):
If new packets arrive and the packet count exceeds the threshold, or the timer expires, the GAM will notify the P again.

Detailed Description Text (1187):
GAM Input Packet Threshold Interrupt (GAB signal: -INTN)

First Hit Fwd Refs

Generate Collection Print

L2: Entry 19 of 29 File: USPT May 27, 1997

US-PAT-NO: 5634015
DOCUMENT-IDENTIFIER: US 5634015 A

TITLE: Generic high bandwidth adapter providing data communications between diverse communication networks and computer system

DATE-ISSUED: May 27, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chang; Paul	Peekskill	NY		
Delp; Gary S.	Yorktown Heights	NY		
Meleis; Hanafy E.-S.	Yorktown Heights	NY		
Montalvo; Rafael M.	Yorktown Heights	NY		
Seidman; David I.	New York	NY		
Tantawy; Ahmed N.-E.-D.	Yorktown Heights	NY		
Zumbo; Dominick A.	White Plains	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
IBM Corporation	Armonk	NY			02

APPL-NO: 08/ 317894 [PALM]
DATE FILED: October 4, 1994

PARENT-CASE:
This is a divisional of application Ser. No. 651,894, filed on Feb. 6,1991, which is now issued U.S. Pat. No. 5,367,643.

INT-CL: [06] G06 F 13/00

US-CL-ISSUED: 395/309; 364/927.92, 364/927.93, 364/927.96, 364/940, 364/940.61, 364/DIG.2, 370/402, 370/412, 395/800
US-CL-CURRENT: 710/310; 370/402, 370/412, 710/62

FIELD-OF-SEARCH: 370/85.1-85.15, 395/250, 395/325, 395/500, 395/600, 395/800, 395/309, 371/32

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5043981	August 1991	Firoozmand et al.	370/85.1
☐	5153884	October 1992	Lucak et al.	371/32
☐	5187780	February 1993	Clark et al.	395/325
☐	5193149	March 1993	Awiszio et al.	395/200
☐	5247626	September 1993	Firoozmand et al.	395/250
☐	5367643	November 1994	Chang et al.	395/325

OTHER PUBLICATIONS

"Five Chip Token-Passing Set Operates LANs At 100 Mbit/s" By Nicolas Mikhoff, Electronic Design, v35, p. 45(5), Sep. 17, 1987.
"VLSI Ethernet Controller Targets Future LANs"; Leonard Milt, Electronic Design, v38, n2, p. 105(3); Jan. 25, 1990.

ART-UNIT: 237

PRIMARY-EXAMINER: Black; Thomas G.

ASSISTANT-EXAMINER: Alam; Hosain T.

ATTY-AGENT-FIRM: Scully, Scott, Murphy and Presser

ABSTRACT:

A generic high bandwidth adapter providing a unified architecture for data communications between buses, channels, processors, switch fabrics and/or communication networks. Data is carried by data stream packets of variable lengths, and each packet includes a header control information portion required by communication protocols used to mediate the information exchange, and normally a data portion for the data which is to be communicated. A packet memory stores data packets arriving at a plurality of generic adapter input/output ports. The packet memory is segmented into a plurality of buffers, and each data packet is stored in one or more buffers as required by the length thereof. A generic adapter manager is provided for performing and synchronizing generic adapter management functions, including implementing data structures in the packet memory by organizing data packets in buffers, and organizing data packets into queues for processing by the processor subsystem or transfer to or from generic adapter input/output ports. The generic adapter manager prepares future response to anticipated requests for communications services which are functions of the current requests for communication services, such as preparing a response for an anticipated request for a next buffer by a current request for a receipt of data. The generic adapter manager stores the future responses at specified addresses in memory which can be read by a requester. Each generic adapter input/output port has associated therewith a packet memory interface providing for the transfer of data packets into and out of the packet memory, such that when a data packet is received at an input/output port, the data packet is transferred into the adapter packet memory and queued for processing.

19 Claims, 13 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L2: Entry 22 of 29 File: USPT Nov 22, 1994

US-PAT-NO: 5367643
DOCUMENT-IDENTIFIER: US 5367643 A
**** See image for Certificate of Correction ****

TITLE: Generic high bandwidth adapter having data packet memory configured in three level hierarchy for temporary storage of variable length data packets

DATE-ISSUED: November 22, 1994

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chang; Paul	Peekskill	NY		
Delp; Gary S.	Yorktown Heights	NY		
Meleis; Hanafy E.	Yorktown Heights	NY		
Montalvo; Rafael M.	Yorktown Heights	NY		
Seidman; David I.	New York	NY		
Tantawy; Ahmed N.	Yorktown Heights	NY		
Zumbo; Dominick A.	White Plains	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			02

APPL-NO: 07/ 651894 [PALM]
DATE FILED: February 6, 1991

INT-CL: [05] G06F 13/00

US-CL-ISSUED: 395/325; 395/200, 395/800, 370/60, 370/85.13, 370/85.14, 370/94.1, 370/94.2, 364/927.92, 364/927.93, 364/927.96, 364/940, 364/940.61, 364/DIG.2
US-CL-CURRENT: 710/62; 370/412, 709/231

FIELD-OF-SEARCH: 395/800, 395/200, 395/325, 395/500, 370/85.13, 370/85.14, 370/94.1, 370/94.2, 370/60, 370/60.1, 370/85.6

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4621362	November 1986	Sy	370/85.14

4878218	October 1989	Takada	370/94.1
4947390	August 1990	Sheehy	370/85.13
5058109	October 1991	Goldberg et al.	370/85.13
5086426	February 1992	Tsukakoshi et al.	370/85.13
5088090	February 1992	Yacoby	370/85.13
5187780	February 1993	Clark et al.	395/325
5193149	March 1993	Awiszio et al	395/200

OTHER PUBLICATIONS

Jin-tu Liu, "Generalized State Machine for Generic Adapter Bus Master Interface," Research Report RC 16106 (#71520) Sep. 14, 1990.
Ramachandran et al., "Hardware Support for Interprocess Communication," IEEE Transactions on Parallel and Distributed Systems, vol. 1, No. 3, Jul. 1990.

ART-UNIT: 237

PRIMARY-EXAMINER: Lee; Thomas C.

ASSISTANT-EXAMINER: Harrity; Paul

ATTY-AGENT-FIRM: Scully, Scott, Murphy & Presser

ABSTRACT:

A generic high bandwidth adapter providing a unified architecture for data communications between buses, channels, processors, switch fabrics and/or communication networks. Data is carried by data stream packets of variable lengths, and each packet includes a header control information portion required by communication protocols used to mediate the information exchange, and normally a data portion for the data which is to be communicated. The generic high bandwidth adapter comprises a processor subsystem including a processor for processing the header control information portions of data packets. The processor has access to data packets stored in a packet memory which stores data packets arriving at four generic adapter input/output ports. The packet memory is segmented into a plurality of buffers, and each data packet is stored in one or more buffers as required by the length thereof. A generic adapter manager is provided for performing and synchronizing generic adapter management functions, including implementing data structures in the packet memory by organizing data packets in buffers, and organizing data packets into queues for processing by the processor subsystem or transfer to or from generic adapter input/output ports. Each generic adapter input/output port has associated therewith a packet memory interface providing for the transfer of data packets into and out of the packet memory, such that when a data packet is received at an input/output port, the data packet is transferred into the adapter packet memory and queued for processing.

24 Claims, 13 Drawing figures

US006467008B1

(12) **United States Patent**
Gentry, Jr. et al.

(10) **Patent No.: US 6,467,008 B1**
(45) **Date of Patent: Oct. 15, 2002**

(54) **METHOD AND APPARATUS FOR INDICATING AN INTERRUPT IN A NETWORK INTERFACE**

(75) Inventors: **Denton E. Gentry, Jr.**, Fremont, CA (US); **Linda T. Cheng**, San Jose, CA (US)

(73) Assignee: **Sun Microsystems, Inc.**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/259,936**

(22) Filed: **Mar. 1, 1999**

(51) **Int. Cl.**^{7} **G06F 13/24**
(52) **U.S. Cl.** **710/261**
(58) **Field of Search** 710/260–269

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,319,752 A		6/1994	Petersen et al.	395/250
5,414,858 A	*	5/1995	Hoffman et al.	340/3.4
5,463,752 A	*	10/1995	Benhase et al.	710/262
5,471,618 A	*	11/1995	Isfeld	340/3.51
5,485,584 A		1/1996	Hausmann et al.	395/842
5,797,037 A		8/1998	Ecclesine	395/868
6,065,073 A	*	5/2000	Booth	370/908

FOREIGN PATENT DOCUMENTS

EP	0 4 96 177 A1	7/1992	G06F/13/28
EP	0 752 799 A1	1/1997	H04Q/11/04
EP	0 852 357 A2	7/1998	G06F/13/24
GB	2 282 474	4/1995	

OTHER PUBLICATIONS

Pending U.S. patent application Ser. No. 09/259,445, entitled "Method and Apparatus for Distributing Network Processing on a Multiprocessor Computer," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3481-JTF).

Pending U.S. patent application Ser. No. 09/260,367, entitled "Method and Apparatus for Suppressing Interrupts in a High-Speed Network Environment", by Denton Gentry, filed Mar. 1, 1999 (Attorney Docket SUN-P3482-JTF).

Pending U.S. patent application Ser. No. 09/259,736, entitled "Method and Apparatus for Modulating Interrupts in a Network Interface," by Denton Gentry et al., filed Mar. 1, 1999 (Attorney Docket SN-P3483-JTF).

Pending U.S. patent application Ser. No. 09/259,765, entitled "A High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3485-JTF).

Pending U.S. patent application Ser. No. 09/260,618, entitled "Method and Apparatus for Classifying Network Traffic in a High Performance Network Interface," by Shimon Muller et al., filed Mar. 1, 1999 (Attorney Docket SUN-P3486-JTF).

(List continued on next page.)

Primary Examiner—Sumati Lefkowitz
(74) *Attorney, Agent, or Firm*—Park, Vaughan & Fleming LLP

(57) **ABSTRACT**

A network interface is polled by a process operating on a host computer system. Each time the network interface is polled the process determines whether any packets have been transferred to the host. If so, they are processed. Interrupts that would normally be issued from the network interface in response to the transfer of packets are suppressed or postponed during the polling mode of operation. If, however, a predetermined period of time has elapsed or a predetermined number of packets have been transferred since a previous poll or a previous interrupt, then an interrupt may be generated.

21 Claims, 7 Drawing Sheets

BEST AVAILABLE COPY

US-PAT-NO: 6467008

DOCUMENT-IDENTIFIER: US 6467008 B1
See image for Certificate of Correction

TITLE: Method and apparatus for indicating an interrupt in a network interface

---------- KWIC ---------

Brief Summary Text - BSTX (12):
In one embodiment of the invention a system and method for polling a network interface are provided. In this embodiment an interrupt that would normally alert a host processor to the arrival of a network packet is suspended during a polling mode of operation. Each time the network interface is polled, any waiting packets are processed. If a threshold amount of time or a threshold number of packets are received without being processed, however, interrupts may be enabled to ensure the packets are serviced. Thus, a polling mode of operation may be combined with interrupt modulation in one embodiment of the invention.

Detailed Description Text - DETX (22):
Illustratively, the time counter and packet counter are set to threshold values after a host computer processes an interrupt. Suitable threshold values, which may be stored in programmable registers or other data structures, are twenty microseconds of time and seven packets for a moderate level of traffic. Thereafter, the time counter decrements (e.g., counts down toward zero) in response to a clock signal or other means of noting the passage of time, and the packet counter decrements in response to a packet transfer signal or other indicator that a packet was transferred to the host computer. A packet transfer signal may, for example, be generated by a NIC module responsible for copying a packet to host memory. Until either of the time counter or packet counter reaches zero or some other final value, which may be stored in a register or other programmable data storage unit, an interrupt will not be generated in response to the transfer of a packet. After a final value is reached, an interrupt may be generated for a packet transferred after the last interrupt was processed by the host computer.

US-PAT-NO: 6216182

DOCUMENT-IDENTIFIER: US 6216182 B1

TITLE: Method and apparatus for serving data with adaptable interrupts

---------- KWIC ---------

Brief Summary Text - BSTX (6):

Other existing methods use a counter or timer to limit the number of interrupts but can have latency problems especially for video or voice data. An interrupt is generated every N packet and M clock ticks from the time the last packet was enqueued. This method attempts to minimize the number of interrupt by generating an interrupt after N packets are enqueued (N is programmable). To minimize latence in case N is large, an interrupt is generated based on a timer. The timer is triggered at the end of a packet. If the timer crosses a M threshold (programmable) without detecting the end of another packet, then an interrupt is generated. Using this scheme requires the programming of two parameters: N and M. It is difficult for the host to determine the optimum value for N and M for different load conditions and variations of the host and card.

Detailed Description Text - DETX (13):

In all the schemes described above, there is an additional condition that can generate an interrupt. An interrupt is generated if the number of buffers 18 pending in the queue 16 reaches a high threshold. This is needed to prevent overflowing the queue 16 in the case of a very large packet (the queue 16 is almost full, but the end of the packet is not yet received).

US006216182B1

(12) **United States Patent**
Nguyen et al.

(10) **Patent No.: US 6,216,182 B1**
(45) **Date of Patent: Apr. 10, 2001**

(54) **METHOD AND APPARATUS FOR SERVING DATA WITH ADAPTABLE INTERRUPTS**

(75) Inventors: **Nhiem Nguyen**, Cranberry; **Michael H. Benson**, New Castle; **Steven J. Schlick**, Wexford; **George Totolos, Jr.**, Cranberry, all of PA (US)

(73) Assignee: **Fore Systems, Inc.**, Warrendale, PA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/126,975**

(22) Filed: **Jul. 30, 1998**

(51) **Int. Cl.**^{7} **G06F 5/00**
(52) **U.S. Cl.** **710/48**; 710/60; 710/33; 710/34; 710/58; 709/233
(58) **Field of Search** 710/48–57, 60, 710/260, 262, 263, 267–268, 129, 128, 62, 63; 709/201, 203, 217, 231, 233, 235

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,068,784 *	11/1991	Kishino et al.	710/100
5,586,294 *	12/1996	Goodwin et al.	711/114
5,630,111 *	5/1997	Yoshida	710/60
5,732,285 *	3/1998	Harrison et al.	710/56
5,740,467 *	4/1998	Chmielecki, Jr. et al.	710/56
5,758,191 *	5/1998	Kasebayashi et al.	710/56
5,778,218 *	7/1998	Gulick	710/38
5,802,153 *	9/1998	Sridhar et al.	379/98
5,809,337 *	9/1998	Hannah et al.	710/33
5,809,338 *	9/1998	Klein	710/35
5,815,677 *	9/1998	Goodrum	710/126
5,842,044 *	11/1998	No	710/53
5,941,962 *	8/1999	Hirano	710/53
5,978,868 *	11/1999	Maas	710/52
5,983,293 *	11/1999	Murakami	710/56
6,138,190 *	10/2000	Nording	710/60

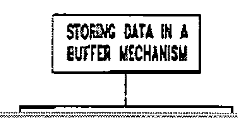
* cited by examiner

Primary Examiner—Robert Beausoleil
Assistant Examiner—Raymond N Phan
(74) *Attorney, Agent, or Firm*—Ansel M. Schwartz

(57) **ABSTRACT**

A system for storing data. The system includes a host for processing the data. The system includes a buffer mechanism for storing data and producing interrupt signals to the host for informing the host there is data in the buffer mechanism for the host to process. The buffer mechanism adapting the production of interrupts based on the speed the host can process data. The host is in contact with the buffer mechanism. A method for serving data. The method includes the steps of storing data in a buffer mechanism. Then there is the step of sending an initial interrupt signal to a host from the buffer mechanism informing the host there is data in the buffer mechanism for the host to process. Next there is the step of transferring data in the buffer mechanism to the host. Then there is the step of processing data from the buffer mechanism with the host. Next there is the step of adapting when a subsequent interrupt signal is sent to the host based on the speed the host can process data. Then there is the step of sending the subsequent interrupt signal to the host from the buffer mechanism when there is data in the buffer mechanism for the host to process.

18 Claims, 2 Drawing Sheets



L3: Entry 2 of 2 File: USPT Feb 10, 1998

DOCUMENT-IDENTIFIER: US 5717932 A
TITLE: Data transfer interrupt pacing

Abstract Text (1):
A communications network adapter of the type coupling a computer, in which the computer includes a microprocessor, main memory and a system bus, that controls host interrupts in a manner to improve system performance. The adapter includes a buffer memory for storing data to be transferred between the bus an the network, and a transfer controller that controls the transfer of data between the main memory and the buffer memory and between the network and the buffer memory. The adapter also includes an interrupt controller that monitors predetermined events relating to data transfer between the computer and the network, and that causes the sending of interrupt signals to the microprocessor. Interrupt signals cause the microprocessor to initiate processing associated with the transfer of data between the computer and the network. According to one aspect of the invention the adapter includes an interrupt pacing timer that prevents the sending of interrupts to the microprocessor from the adapter for predetermined time after an interrupt acknowledgement signal is received from the microprocessor. According to another aspect of the invention an interrupt threshold counter is provided that prevents the sending of interrupts to the microprocessor until a predetermined plurality of frames are transmitted.

Brief Summary Text (11):
In accordance with the present invention, a communications network adapter is provided, coupling a computer, wherein the computer includes a microprocessor, main memory and a system bus interconnecting the microprocessor and the main memory, to a network. The adapter includes a buffer memory for storing data intended for transfer between the bus and the network, and a transfer controller that controls the transfer of data between the main memory and the buffer memory by way of the bus, and that controls the transfer of data between the network and the buffer memory. An interrupt controller is also provided, that monitors predetermined events related to data transfer between the computer and the network and causes the sending of interrupt signals to the microprocessor. The interrupt signals cause the microprocessor to initiate processing associated with the transfer of data between the computer and network. According to one aspect of the invention an interrupt pacing timer is also provided that prevents the sending of interrupts to the microprocessor for a predetermined time period after an interrupt acknowledgement signal is received from the microprocessor. In accordance with another aspect of the invention an interrupt threshold counter is provided that prevents the sending of interrupts to the microprocessor until a predetermined plurality of frames are transmitted.

Detailed Description Text (6):
FIG. 4 is a high level block diagram similar to that shown in FIG. 2, showing a data transfer state machine 10, interrupt block 12 and elements associated with the data transfer 14, as before. However, newly presented in accordance with the preferred embodiment is a transmit threshold counter 32 and a pacing timer 34, as shown. The transmit threshold counter 32 monitors the number of frames that have

been successfully completed, and only after a predetermined number of such frames have been transmitted does it communicate with the interrupt block 12 to cause the sending of an interrupt associated with such transmissions. Pacing timer 34 monitors all events giving rise to an interrupt and allows interrupts to be communicated to the CPU 1 only after a predetermined time interval after the successful servicing of the previous interrupt.

First Hit Fwd Refs

End of Result Set

Generate Collection Print

L3: Entry 2 of 2 File: USPT Feb 10, 1998

US-PAT-NO: 5717932

DOCUMENT-IDENTIFIER: US 5717932 A

TITLE: Data transfer interrupt pacing

DATE-ISSUED: February 10, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Szczepanek; Andre	Bedford			GB2
Beaudoin; Denis R.	Missouri City	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Texas Instruments Incorporated	Dallas	TX			02

APPL-NO: 08/ 334511 [PALM]

DATE FILED: November 4, 1994

INT-CL: [06] G06 F 13/00

US-CL-ISSUED: 395/733; 395/557, 395/309

US-CL-CURRENT: 710/260; 713/502

FIELD-OF-SEARCH: 395/297, 395/304, 395/865, 395/733, 395/734, 395/735, 395/736, 395/557, 395/308, 370/85.1

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3851104	November 1974	Willard et al.	370/85.1
☐	4023143	May 1977	Braunstein	395/736
☐	4099255	July 1978	Stanley et al.	395/733
☐	4161786	July 1979	Hopkins et al.	395/297
☐	4218739	August 1980	Negi et al.	395/735
☐	4481572	November 1984	Ochsner	395/297

☐	4823312	April 1989	Michael et al.	395/250
☐	4935894	June 1990	Ternes et al.	395/308
☐	4969120	November 1990	Azevedo et al.	395/297
☐	4979100	December 1990	Makris et al.	395/297
☐	5032982	July 1991	Dalrymple et al.	395/350
☐	5165032	November 1992	Herbault	395/200.06
☐	5255373	October 1993	Brockmann et al.	395/325
☐	5287458	February 1994	Michael et al.	395/250
☐	5307459	April 1994	Peterson et al.	395/200.03
☐	5377332	December 1994	Entwistle	395/297
☐	5386573	January 1995	Okamoto	395/733
☐	5392435	February 1995	Mashi et al.	395/733
☐	5423049	June 1995	Kurihara	395/733
☐	5469543	November 1995	Nishihara et al.	395/200.06
☐	5475816	December 1995	Yonezawa et al.	395/200.06
☐	5506975	April 1996	Onodera	395/733
☐	5506993	April 1996	Fitch et al.	395/735

ART-UNIT: 235

PRIMARY-EXAMINER: Sheikh; Ayaz R.

ATTY-AGENT-FIRM: Moore; J. Dennis Kesterson; James C. Donaldson; Richard L.

ABSTRACT:

A communications network adapter of the type coupling a computer, in which the computer includes a microprocessor, main memory and a system bus, that controls host interrupts in a manner to improve system performance. The adapter includes a buffer memory for storing data to be transferred between the bus an the network, and a transfer controller that controls the transfer of data between the main memory and the buffer memory and between the network and the buffer memory. The adapter also includes an interrupt controller that monitors predetermined events relating to data transfer between the computer and the network, and that causes the sending of interrupt signals to the microprocessor. Interrupt signals cause the microprocessor to initiate processing associated with the transfer of data between the computer and the network. According to one aspect of the invention the adapter includes an interrupt pacing timer that prevents the sending of interrupts to the microprocessor from the adapter for predetermined time after an interrupt acknowledgement signal is received from the microprocessor. According to another aspect of the invention an interrupt threshold counter is provided that prevents the sending of interrupts to the microprocessor until a predetermined plurality of frames are transmitted.

9 Claims, 6 Drawing figures

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 5 of 5 returned.

☐ 1. Document ID: US 6467008 B1
L4: Entry 1 of 5 File: USPT Oct 15, 2002

US-PAT-NO: 6467008
DOCUMENT-IDENTIFIER: US 6467008 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for indicating an interrupt in a network interface

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 2. Document ID: US 6434651 B1
L4: Entry 2 of 5 File: USPT Aug 13, 2002

US-PAT-NO: 6434651
DOCUMENT-IDENTIFIER: US 6434651 B1

TITLE: Method and apparatus for suppressing interrupts in a high-speed network environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 3. Document ID: US 6216182 B1
L4: Entry 3 of 5 File: USPT Apr 10, 2001

US-PAT-NO: 6216182
DOCUMENT-IDENTIFIER: US 6216182 B1

TITLE: Method and apparatus for serving data with adaptable interrupts

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☐ 4. Document ID: US 5634015 A
L4: Entry 4 of 5 File: USPT May 27, 1997

US-PAT-NO: 5634015

DOCUMENT-IDENTIFIER: US 5634015 A

TITLE: Generic high bandwidth adapter providing data communications between diverse communication networks and computer system

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 5. Document ID: US 5367643 A

L4: Entry 5 of 5 File: USPT Nov 22, 1994

US-PAT-NO: 5367643
DOCUMENT-IDENTIFIER: US 5367643 A
**** See image for Certificate of Correction ****

TITLE: Generic high bandwidth adapter having data packet memory configured in three level hierarchy for temporary storage of variable length data packets

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L1 and L3	5

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 5 of 5 returned.

☐ 1. Document ID: US 6467008 B1

L4: Entry 1 of 5 File: USPT Oct 15, 2002

US-PAT-NO: 6467008
DOCUMENT-IDENTIFIER: US 6467008 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for indicating an interrupt in a network interface

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 2. Document ID: US 6434651 B1

L4: Entry 2 of 5 File: USPT Aug 13, 2002

US-PAT-NO: 6434651
DOCUMENT-IDENTIFIER: US 6434651 B1

TITLE: Method and apparatus for suppressing interrupts in a high-speed network environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 3. Document ID: US 6216182 B1

L4: Entry 3 of 5 File: USPT Apr 10, 2001

US-PAT-NO: 6216182
DOCUMENT-IDENTIFIER: US 6216182 B1

TITLE: Method and apparatus for serving data with adaptable interrupts

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 4. Document ID: US 5634015 A

L4: Entry 4 of 5 File: USPT May 27, 1997

US-PAT-NO: 5634015

DOCUMENT-IDENTIFIER: US 5634015 A

TITLE: Generic high bandwidth adapter providing data communications between diverse communication networks and computer system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 5. Document ID: US 5367643 A

L4: Entry 5 of 5 File: USPT Nov 22, 1994

US-PAT-NO: 5367643
DOCUMENT-IDENTIFIER: US 5367643 A
**** See image for Certificate of Correction ****

TITLE: Generic high bandwidth adapter having data packet memory configured in three level hierarchy for temporary storage of variable length data packets

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L1 and L3	5

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

h e b b g e e e f e f ef b e